

3D hahmon mallinnus ja ohjaaminen pelimoottorilla

Milla Kiviluoma

Tekijä(t) Milla Kiviluoma	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Opinnäytetyön otsikko 3D hahmon mallinnus ja ohjaaminen pelimoottorilla	Sivu- ja liitesivumäärä 43
<p>Tämän opinnäytetyön tarkoituksena on tutkia 3D grafiikan historiaa, käyttökohteita sekä tulevaisuutta. 3D grafiikan sisältämistä aloista tutkittiin erityisesti 3D-mallinnusta sekä animointia. Tavoitteena on oppia miten 3D grafiikka on kehittynyt ja mitä mahdollisuuksia löytyy sen tulevaisuudesta. Tämän lisäksi opinnäytetyössä vertaillaan eri 3D-mallinnus ohjelmia sekä pelimoottoreita. Tavoitteena on oppia miten erityisesti tuloksessa käytetty Blender 3D-mallinnus ohjelmistoa ja Unity pelimoottoria voidaan käyttää yhdessä ja miksi juuri näitä ohjelmistoa käytetään tuloksen luonnissa.</p> <p>3D-mallinnuksen ohjelmat ovat tärkeä osa aiheen tutkimista. Kyseisiä ohjelmistoa on listattu ja vertailtu, ja tuloksessa käytetty Blender ohjelmistoa tutkitaan syvällisemmin. Pelimoottoreita on samoin vertailtu, ja Unity pelimoottoriin perehdytään paremmin. Tuloksessa käytetyt ohjelmistot ovat suosittuja, sekä suurimmaksi osaksi ilmaisia joka mahdollistaa käytön kaikille kiinnostuneille. 3D animaatiota videopeleissä tutkittiin, jotta voidaan ymmärtää tuloksen tarkoitus.</p> <p>Tuloksen syntymisen tärkeimpiä tekniikoita tutkitaan. 3D animoidun hahmon valmistaminen animointiin sekä pelimoottorissa käyttöön vaatii luurankoa ja sen liikutettavuutta. Tällaisen hahmon luominen vaatii rigaus tekniikkaa sekä käänteisen kinematiikan tekniikkaa.</p> <p>Edellä mainittujen aiheiden tutkimisen tarkoitus on luoda ymmärrettävä kokonaisuus tulokseen johtavista ohjelmistoista ja tekniikoista. Opinnäytetyössä tuotettiin 3D-mallinnettu hahmo pelikäyttöön. Työ tehtiin projektiluontoisesti, jossa tuloksena syntyi lyhyt video hahmon liikkeistä. Hahmo mallinnettiin Blender ohjelmistolla huolehtien siitä, että hahmo soveltuu käyttöön Unity pelimoottorissa. Mallinnuksen lisäksi lisättiin hahmolle materiaaleja ja luuranko, jota käytettiin animoinnin luomiseen. Valmis malli tuotiin pelimoottoriin, jossa sen luotuja ominaisuuksia käytettiin hahmon liikuttamiseen näppäimistöllä.</p> <p>Tuloksena syntyi hahmo joka on valmis käytettäväksi peliprojekteissa. Lisäksi tuotettiin lyhyt video kuvastamaan hahmon kävelyä. Blenderin ja Unityn käyttö yhdessä onnistuu erinomaisesti, sillä ne tukee toisiaan. 3D grafiikan ala jatkaa kasvuaan ja sen tulevaisuus on varmasti suuri.</p>	
Asiasanat 3D-mallinnus, animaatio, Blender, Unity, kinematiikka	

Sisällys

1	Johdanto	1
1.1	Rakenne	1
1.2	Sanasto.....	2
2	3D grafiikka	3
2.1	Historia	3
2.2	Käyttökohteet.....	4
2.3	Tulevaisuus.....	4
3	Mallinnus ja animaatio.....	4
3.1	3D-mallinnus ohjelmistot.....	5
3.2	Pelimootorit.....	6
3.3	3D animaatio videopeleissä	7
4	Tekniikat.....	8
4.1	Rigaus tekniikka.....	8
4.2	Inverse kinematics tekniikka.....	8
5	Blender	9
5.1	Historia	9
5.2	Käyttö	9
6	Unity.....	12
6.1	Historia	12
6.2	Käyttö	12
7	Hahmon mallinnus	15
7.1	Suunnittelu.....	15
7.2	Toteutus.....	15
7.2.1	Hahmon luominen.....	15
7.2.2	Rigaus.....	20
7.2.3	Animointi	27
8	Hahmon liikuttaminen pelimootori Unitylla	28
8.1	Hahmon importoiminen Unityyn	28
8.2	Animaation asettaminen hahmolle	31
8.3	Hahmon pelillistäminen	33
8.4	Tulos.....	37
9	Pohdinta.....	38
	Lähteet	40

1 Johdanto

Nykyaikana on lähes mahdotonta elää päivän näkemättä jonkinlaista 3D grafiikkaa. 3D grafiikka on löytänyt tiensä lähes kaikkiin aloihin, ja sen käyttö on jokapäiväistä. Yksi näkyvimmistä 3D aloista on videopeliteollisuus. Nykyajan videopelit ovat lisääntyvissä määrin luotuja 3D tekniikoita käyttäen, muuttuen aina vain realistisemmiksi. Kiinnostus alaan on suurta ja se jatkaa kasvuaan.

Tämän opinnäytetyön tarkoituksena on tutkia 3D grafiikan historiaa, käyttökohteita sekä tulevaisuutta. 3D grafiikan sisältämistä aloista tutkittiin erityisesti 3D-mallinnusta sekä animointia. Tavoitteena on oppia miten 3D grafiikka on kehittynyt ja mitä mahdollisuuksia löytyy sen tulevaisuudesta. Tämän lisäksi opinnäytetyössä vertaillaan eri 3D-mallinnus ohjelmia sekä pelimoottoreita. Tavoitteena on oppia miten erityisesti tuloksessa käytettyä Blender 3D-mallinnus ohjelmistoa ja Unity pelimoottoria voidaan käyttää yhdessä ja miksi juuri näitä ohjelmistoja käytetään tuloksen luonnissa.

3D-mallinnuksen ohjelmat ovat tärkeä osa aiheen tutkimista. Kyseisiä ohjelmistoja on listattu ja vertailtu, ja tuloksessa käytettyä Blender ohjelmistoa tutkitaan syvällisemmin. Pelimoottoreita on samoin vertailtu, ja Unity pelimoottoriin perehdytään paremmin. Tuloksessa käytetyt ohjelmistot ovat suosittuja, sekä suurimmaksi osaksi ilmaisia joka mahdollistaa käytön kaikille kiinnostuneille. 3D animaatiota videopeleissä tutkittiin, jotta voidaan ymmärtää tuloksen tarkoitus.

Edellä mainittujen aiheiden tutkimisen tarkoitus on luoda ymmärrettävä kokonaisuus tulokseen johtavista ohjelmistoista ja tekniikoista. Opinnäytetyössä tuotettiin 3D-mallinnettu hahmo pelikäyttöön. Työ tehtiin projektiluontoisesti, jossa tuloksena syntyi lyhyt video hahmon liikkeistä. Hahmo mallinnettiin Blender ohjelmistolla huolehtien siitä, että hahmo soveltuu käyttöön Unity pelimoottorissa. Mallinnuksen lisäksi lisättiin hahmolle materiaaleja ja luuranko, jota käytettiin animoinnin luomiseen. Valmis malli tuotiin pelimoottoriin, jossa sen luotuja ominaisuuksia käytettiin hahmon liikuttamiseen näppäimistöllä.

1.1 Rakenne

Luvussa kaksi tutkitaan 3D grafiikan historiaa, käyttökohteita sekä tulevaisuutta. 3D grafiikan historia on yllättävän pitkä, mutta voidaan ajatella ettei alkuperäinen grafiikka vastannut nykyajan tuotoksia ja mahdollisuuksia juuri ollenkaan. 3D grafiikan käyttöä esiintyy lähes kaikissa aloissa, tulevaisuudessa lisääntyvin määrin.

Luvussa kolme syvennyttään 3D-mallinnukseen sekä animointiin. Vertaillaan ohjelmistoja joilla voidaan tuottaa monia erilaisia 3D projekteja, aina hyvin yksinkertaisista todellisiin taideteoksiin ja elokuvaan saakka. 3D animaatio tulee näkyviin hyvin elokuvissa ja animoiduissa sarjoissa, mutta hyvin suuri animaation alue on videopelituotanto. Vertaillaan pelimoottoreita, painottuen ilmaisiin ohjelmistoihin, jotka mahdollistavat pelien teon kaikille.

Luvussa neljä syvennyttään muutamiin tekniikoihin, jotka ovat välttämättömiä 3D-animaatioon ja joita väistämättä tulee käyttää 3D videopeleissä. Luvussa viisi syvennyttään Blender 3D-mallinnus ohjelmaan, tutkitaan sen historiaa ja sen käytön perusteisiin. Tämän lisäksi tutkitaan Unity ohjelmistoa pelimoottorin kannalta. Miten Unity on kehittynyt sen kehityksestä tähän hetkeen saakka. Unityn käytön perusteet selvitetään alkeellisesti.

Luvussa seitsemän aloitetaan hahmon mallinnus Blenderissä varmistaen että se soveltuu pelimoottorissa käyttöön. Hahmolle luodaan luuranko ja kävelysykli. Luvussa kahdeksan tuomme luodun hahmon pelimoottoriin ja hahmo pelillistetään.

1.2 Sanasto

Rigaus	Prosessi, jossa luodaan 3D mallille luuranko, jotta se voi liikkua.
Renderöinti	Tietokoneohjelman avulla tuotettu kuva. Käytetään erityisesti 3D grafiikassa.
Skene	Kuten elokuvan kohtaus. Skene tarkoittaa paikkaa jossa tapahtumat tapahtuvat.
Importointi	Tiedostotyyppin tuominen ohjelmaan, jota ei suoraan tueta kohde ohjelmassa.
Raami	Animaation kuvaruutu. Yhteen sekuntiin voi lisätä useita raameja, jotka ajettuna kokonaisuudessaan näyttävät hahmon liikkeen.

2 3D grafiikka

Kolmiulotteinen grafiikka on matematiikkaa. 3D objekti esiintyy kolmella akselilla, X, Y sekä Z akseleilla. Kaksiulotteiset grafiikat voidaan luoda käyttämällä X akselia, joka on horisontaalinen ja Y akselia, joka on vertikaalinen. 3D grafiikassa käytetään edellä mainittujen lisäksi Z akselia, joka kuvastaa syvyyttä. (Slick 2017.)

2.1 Historia

Harvemmin voidaan sanoa teekannun olevan se tärkein objekti osana jotain suurta, tietokonegrafiikassa voidaan kuitenkin näin sanoa. Tämän teekannun nimi on "Utah teapot", eli "Utahin teekannu". Teekannu sai alkunsa 1974, jolloin Martin Newell yritti keksiä hyviä ideoita algoritmeihin, joilla voidaan realistisesti näyttää 3D muotoja. Teekannu oli täydellinen objekti tietokonegrafiikan tutkimiseen, sillä sen muodot ovat monipuolisemmat kuin useat muut esineet. Nopeasti teekannusta tuli rakastettu esine grafiikka yhteisössä, vielä tänäkin päivänä teekannu esiintyy useissa elokuvissa, sarjoissa, ja videopeleissä. (Dunietz 2016.)

Tietokonegrafiikka kuitenkin sai alkunsa jo 1960-luvulla. Insinööriopiskelija Ivan Sutherland on usein nimetty tietokonegrafiikan isäksi, Ivan kehitti Sketchpad ohjelman, jolla voitiin piirtää objekteja tietokonenäytöllä sekä tallentaa ne myöhempään käyttöön. (3D Horse 2017.) Sutherland avasi ensimmäisen tietokoneteknologia osaston yhdessä kollega David Evansin kanssa Utahin yliopistossa. Useat menestyneet grafiikkatoimialan henkilöt olivat Sutherlandin opiskelijoita. Sutherlands ja Evans perustivat ensimmäisen 3D grafiikka yrityksen jo vuonna 1969 nimeltään "Evans & Sutherland". (ArchiCGI 2016.)

Tietokonegrafiikka ja siten 3D grafiikka eteni huikaaa vauhtia seuraavat vuosikymmenet. 1990-luvulla tietokonegrafiikasta oli kehittynyt jotain mielenkiintoista ja uskaliaista. Animaatio muuttui realistisemmaksi ja siten mahdollisti ensimmäiset tietokoneella luodut animoidut pitkäkestoiset elokuvat, kuten Toy Story. 3D grafiikkaa aloitettiin käyttää paljon viihde- ja videopeli-alojen lisäksi markkinoinnissa ja teollisuuden automatisoinnissa. (3D Horse 2017.)

2000-luvusta eteenpäin 3D grafiikka on jokapäiväinen ilmiö. Lähestulkoon kaikki teollisuudenalat käyttävät jollain tasolla 3D mallintamista tai suunnittelemista. Viihdealalla tämä vuosikymmen keskittyi tekemään grafiikoita jotka ovat uskottavia, tämä näkyy laskettomista elokuvista ja videopeleistä, jotka kehitettiin tämän mullistuksen aikana. Nykyään 3D mallien teko on paljon nopeampaa ja helpompaa kuin vain muutama

vuosikymmen sitten, kehityksen alla on aina vain realistisemmat grafiikat. (3D Horse 2017.)

2.2 Käyttökohteet

Tunnetuin 3D-mallinnuksen käyttökohde on peliteollisuus. Viime vuosien aikana videopelit ovat muuttuneet realistisemmiksi, maisemat sekä hahmot näyttävät lähes oikeilta. Peliteollisuuden lisäksi suurimmat teollisuusalat jotka käyttävät 3D-mallinnusta ovat muun muassa viihdeteollisuus, arkkitehtuuri, markkinointi sekä tieteenalat. Viihdeteollisuudessa 3D-mallinnus on näkynyt jo jonkin aikaa, oikeiden maisemien ja laitteiden sekaan on usein lisätty 3D-mallinnettuja objekteja. (CAD/CAM Services 2018.)

Viihdeteollisuuden lisäksi useat alat hyödyntävät 3D mallinnusta. Yksi merkittävimmistä aloista jotka käyttävät paljon 3D mallinnusta on lääketieteen ala. Lähes kaikissa lääketieteen osa-alueissa käytetään 3D mallinnusta. Kirurgiopiskelijat harjoittelevat taitojaan ja opiskelevat tuntemaan elimet 3D mallinnetuilla elimillä. Proteeseja tehdään nykyään myös paljon 3D malleilla sekä 3D tulostuksella. (3D Horse 2017.)

Erityisen tuottoisiin teollisuudenaloihin jotka hyödyntävät 3D mallinnusta ovat autoteollisuus, rakennusalat, sekä avaruustutkimus. Kaikki edellämainitut alat käyttävät paljon 3D tulostusta toiminnan tukena, ja 3D tulostuksen hyödyntämistä enemmän ja paremmin tutkitaan yhä enemmän. (3D Horse 2017.)

2.3 Tulevaisuus

Viimeisimmän vuosikymmenen aikana 3D grafiikka on muuttunut erityisen paljon, tulevaisuudessa, se tulee muuttumaan aina enemmän. Suurin tavoite kolmiulotteisessa grafiikassa on luoda fotorealistisia kuvia, ja jotta tähän pystyttäisiin, tulee tietokoneiden laitteiston ja ohjelmistojen kehittyä. On mahdollista, että seuraavien muutamien vuosikymmenien jälkeen, on vaikeaa tunnistaa oikeiden objektien ja 3D grafiikalla luotujen objektien ero. Tulevaisuus alalla on kirkas, haasteet tulevat oikeiden taitojen yhdistämisestä ja teknologian kehittymisestä. Kolmiulotteinen grafiikka tulee muuttamaan maailman lisätystä todellisuudesta todelliseen virtuaalitodellisuuteen. (3D Horse 2017.)

3 Mallinnus ja animaatio

3D-mallinnus on prosessi, jossa luodaan kolmiulotteinen malli mistä tahansa pinnasta tai objektista 3D tilassa. Mallinnus voidaan luoda monilla eri 3D-mallinnus ohjelmistoilla, useilla eri tuotannon aloilla. (Slick 2018.)

3D animaatio videopeleissä eroaa suuresti animoiduista elokuvista. Videopeli animaatio vaatii nopealla aikataululla luotuja monimutkaisia syklejä ja liikkeitä sekä maisemia. Videopeleissä animaation tulee renderöidä ajantasaisesti, heti kun pelaaja kääntää kameran osoittamaan jotakin. Elokuva animointi on videopeli animointia edellä, teknologiat edistyvät hyvin nopeasti ja videopeli animointi saattaa saavuttaa elokuvia vuosien saatossa, viimeisien vuosikymmenien aikana, animaatiot ovat kulkeneet nopeasti toistensa perässä ja edistyneet valtavasti. (Forbes 2016.)

3.1 3D-mallinnus ohjelmistot

3D-mallinnus ohjelmistoja on valtavasti. Useat menestyksekkäimmät sovellukset ovat kalliita ja niitä käytetään suuremmissa tuotannoissa. Useimmat kaupalliset sovellukset tarjoavat ilmaisia kokeiludemoja, mutta kokonaisen lisenssin ostaminen on kallista. Ilmaiset sovellukset ovat alkaneet kuitenkin saavuttaa paljon suosiota sekä harrastelijoiden sekä ammattilaisten keskuudessa. (Slick 2017.)

Ohjelmistoja voidaan lajitella moniin eri tarkoituksiin, jotkut sovellukset ovat parempia perus 3D-mallinnukseen, kun taas toiset parempia esimerkiksi 3D-tulostusta ajatellen. Lisäksi toiset sovellukset soveltuvat parhaiten aloittelijoiden käyttöön helpoimmilla työkaluilla, toiset taas kokeneelle mallintajalle. (Lacoma 2017.)

Ensimmäisenä ohjelmistona usein mainitaan Blender. Blender on ilmainen sovellus joka soveltuu sekä aloittelijoille että kokeneille mallintajille. Blender ohjelmistoa käytetään tämän opinnäytetyön tuloksen luonnissa ja ohjelmistosta kerrotaan paremmin luvussa 5. Seuraavana suosittuna ohjelmistona on Cinema 4D. Cinema 4D vaatii lisenssin, joten ohjelmisto on usein liian kallis harrastajalle. Ohjelmisto on suosittu mainostoimistoissa ja markkinointiin liittyvissä projekteissa, sillä sen toiminnot ovat pääasiallisesti yksityiskohtaisen 3D-mallinnuksen parissa. Ohjelmisto ei taivu kaikkeen, joten se ei sovellu kaikille. Zbrush on hieman halvempi ohjelmisto, perinteisesti käytettynä veistämistyökalu. (Lacoma 2017.)

Seuraava ohjelmisto on Maya. Maya taipuu erinomaisiin yksityiskohtiin, ja se on ideaalinen yksityiskohtaiseen mallinnukseen sekä animointiin. Samantyyppinen ohjelmisto on 3DS Max. Kyseistä ohjelmistoa on käytetty useissa suosituissa videopeleissä sekä muissa huippuprojekteissa. 3DS Max keskittyy yksityiskohtaiseen mallinnukseen ja animointiin sekä renderöintiin. (Lacoma 2017.)

Seuraavat ohjelmistot ovat käytössä enemmän tekniikka alalla, arkkitehtuurissa, ja tuote suunnittelussa. Ensimmäisenä AutoCAD. Ohjelmisto ei juurikaan sovellu graafiseen suunnitteluun ja taiteisiin, se on luotu teknilliseen suunnitteluun. SketchUp on samankaltainen ohjelmisto. Tämä ohjelmisto soveltuu niin aloittelijoille kuin kokeneille käyttäjille. Ohjelmistoa käytetään lähinnä arkkitehtuurisessa suunnittelussa, sisustussuunnittelussa, ja kotisuunnittelussa. (Lacoma 2017.)

3.2 Pelimoottorit

Pelimoottorit ovat ohjelmistoja, joilla voidaan kehittää ja kontrolloida pelejä kunnolla. Pelimoottorien käyttö vaatii paljon harjoittelua ja taitoja, mutta useimmat sisältävät yksinkertaisempia, valmiita moduuleja, kirjastoja, efektejä, ja työkaluja, jotka tekevät työn helpommaksi jo aloittelijoillekin. Pelimoottoreilla voidaan luoda videopelejä tietokoneille, konsoleille sekä mobiililaitteille, jotkut ohjelmistot tosin ovat suunnattu tiettyyn kehitykseen. (Elhady 2017.)

Yksi tunnetuimmista pelimoottoreista on Unreal Engine. Se on jopa Guinness maailman ennätysten mukaan menestyksekkäin pelimoottori. Unreal Engine soveltuu parhaiten suurien monimutkaisten pelien kehittämiseen, sillä se on vahva 3D pelimoottori. Kehitetyn pelin koko on aika suuri ja vaatii erinomaiset laitteet jotta sitä voi pelata. Toinen tunnettu ja laajasti käytetty pelimoottori on Unity. Unity on erinomainen työkalu pelin kehityksen jokaiseen vaiheeseen. Edellä mainittu Unreal Engine pelimoottoria käytetään suurimmaksi osaksi tietokone- ja konsolipelien kehittämiseen, kun taas Unityä käytetään erittäin paljon mobiilipelien sekä virtuaalidellisuus pelien luontiin. (Elhady 2017.)

Seuraava mainitsemisen arvoinen pelimoottori on CryEngine. CryEngine on täysin ilmainen, eikä vaadi rojalteja kehitetyistä peleistä, toisin kuin edellä mainitut pelimoottorit. CryEngine erottuu muista sen erinomaisella grafiikalla sekä pelimoottoriin liitetyn peliaudio työkalun vuoksi. (Elhady 2017.) Kuten aiemmin mainitut pelimoottorit, CryEngine pelimoottorin käyttöön löytyy paljon ohjeita ja kursseja, joita opiskellen saa haltuunsa erinomaisia taitoja pelimoottorien käyttöön. (GameDesigning 2018.) Useissa pelimoottori listauksissa esillä on myös Godot Engine. Godot soveltuu erinomaisesti sekä 2D että 3D pelien luontiin, ja se tarjoaa paljon työkaluja ystävällisellä käyttöliittymällä, joten pelien luonti sujuu artisteilta, suunnittelijoilta, ja animaattoreilta. Samankaltaisuuksista CryEngine pelimoottorin kanssa, voidaan mainita sen ilmaisuus sekä rojaltejen puuttuminen, kaikki kehitetyt tuotteet ovat täysin kehittäjän omaisuutta. (Elhady 2017.)

Pelimoottoreita tutkiessa törmää muutamiin ei niin tunnettuihin moottoreihin jotka ovat kuitenkin erittäin mielenkiintoisia. Yksi esimerkki tällaisesta pelimoottorista on GameMaker. Toisin kuin useimmat pelimoottorit, GameMaker ei vaadi ollenkaan ohjelmointi taitoja. GameMaker mahdollistaa pelien kehityksen kaikille kiinnostuneille, ilman että heidän tulisi opiskella ohjelmointia. Ohjelmointia osaavat kehittäjät saavat myös paljon irti GameMaker:ista, tosin pelimoottorin toiminnallisuudet ovat paljon rajallisempia kuin toisissa. (GameDesigning 2018.)

3.3 3D animaatio videopeleissä

Animaation voidaan kuvitella olevan hyvin samanlaista videopeleissä sekä elokuvissa, mutta tämä ei ole totta. Animaatioihin voidaan käyttää samoja työkaluja mutta prosessit ja tekniikat eroavat toisistaan valtavasti. (Pluralsight 2014.)

Suurin ero elokuvien ja videopelien animoimisessa tulee kameroista, mihin kamera osoittaa. Elokuvien animoinnissa on vain yksi kamera, joka osoittaa aina tiettyyn kohtaukseen. Elokuvan animoijan tulee animoida vain mitä kamera näkee. Videopelien animoinnissa tulee ottaa huomioon kaikki ympärillä oleva. Pelaaja on useimmiten täysin kontrollissa hahmosta ja siitä mihin kamera osoittaa, täten animaation tulee näyttää hyvältä jokaisesta suunnasta ja kulmasta. (Pluralsight 2014.)

Videopelejä animoidessa voi animoida hyvin samankaltaisia hahmoja ja maisemia kuin animoiduissa elokuvissa ja vaihtoehtoisesti animoida erinomaisia realistisia kohtauksia. Videopeli animaattorin työhön kuuluu hahmojen liikkeiden, taistelukohtausten, välianimaatioiden, ja useiden muiden kohtausten animointi. Animaattorin on aina muistettava huomioida kolmiulotteisuus ja varmistaa että animoidut liikkeet ja maisemat ovat näyttäviä ja hyvä laatuista joka suunnasta. Liikkeet saattavat olla yksinkertaisia, kuten pelkän hengitys syklin tai seisomis syklin luominen, kaikki tämä kuitenkin vaatii paljon työtä. Animaatioiden ajoitus riippuu pelaajan syötteestä, kun pelaaja painaa tiettyä nappia, tulee jotakin tapahtua, animaation kesto tulee olla sopiva kohtaukseen. Videopelien animoinnissa tulee aina huomioida mitä pelaaja saattaa tehdä ja missä tilanteissa. Yksi suuri eroavaisuus elokuva ja videopeli animaatioissa on se, kuinka kauan animaattorilla on aikaa valmistaa animaatio. Elokuva animaattorilla saattaa olla kuukausia aikaa valmistaa yksi kohtaus, tavallaan siis yksi kuva. Videopeli animaattorilla aikaa ei ole niin paljon, moderneissa videopeleissä on paljon hahmoja, joka tarkoittaa paljon animaatioita. Videopeli animaattorin tulee saavuttaa erinomaisia animaatioita lyhyessä ajassa, muistaen kaikki kuvakulmat. Tähän on kuitenkin tullut apuna liikkeen tunnistus (motion capture) tekniikka. Liikkeen tunnistus tekniikka toimii siten, että oikea henkilö

pukee päällensä liikkeen tunnistus puvun, henkilö tekee liikkeitä tai näyttelee kohtauksia, jotka kuvataan ja liikkeet tunnistetaan koneella. Tunnistetut liikkeet voidaan sitten liittää mallinnettuun hahmoon ja tällä tavalla voidaan luoda erittäin uskottavia liikeanimaatioita. (Pluralsight 2014.)

4 Tekniikat

Animointi tekniikoita on useita riippuen hyvinkin siitä mihin käyttötarkoitukseen animoidaan. Perinteinen animointi on kaksiulotteisten kuvien piirtäminen ja siten animaation luonti. Kaksiulotteista animaatiota voidaan myös luoda samoin kuin kolmiulotteisia animaatioita, jättäen vain pois animaation syvyyden. Ensimmäiset videopelit olivat kaksiulotteisia eikä tekniikka ole hävinnyt vaikka kolmiulotteinen animaatio onkin kerännyt suosiota. 3D animaatiossa on paljon samanlaisuuksia stop motion animaation kanssa. Molemmissa hahmojen asentoja muutetaan raami raamilta, mutta 3D animaatio tapahtuu digitaalisessa työtilassa ja on siten hallittavampaa. Piirtämisen tai fyysisen objektin liikuttamisen sijaan hahmot mallinnetaan ohjelmistossa ja sille asetetaan luuranko. Luuranko mahdollistaa hahmon liikuteltavuuden eri asentoihin. (Bloop 2018.) Tässä opinnäytetyössä käytetään erityisesti rigaus tekniikkaa ja käänteistä kinematiikkaa, jotka liittyvät hahmon luurankoon ja sen animointiin.

4.1 Rigaus tekniikka

Hahmon animointi vaatii sen, että hahmolla on jotain mitä liikuttaa. Pelkän hahmon luominen luo patsaan, jota on mahdoton liikuttaa ilman tiettyä köysistöä. Tämä vaatii rigaus tekniikkaa, "rigging" englanniksi. Hahmon "rig" on periaatteessa sen luuranko, joka on liitetty hahmo objektiin. Luuranko koostuu nivelistä ja luista, joita käytetään kun animoidaan hahmon liikkeitä. Ensimmäinen nivel tulisi olla hahmon juuri, se kohta josta kaikki muut luut ja nivelet lähtevät. Nivelet usein vaativat myös rajoituksia, jotta animointi olisi mahdollisimman realistista. Hahmon kasvojen rigaus on usein erillään muusta luurangosta ja vaatii paljon yksityiskohtia. (Slick 2018.)

4.2 Inverse kinematics tekniikka

Animoinnin kinematiikka tekniikoita ovat Inverse Kinematics eli Käänteinen kinematiikka, sekä Forward Kinematics eli Eteenpäinen kinematiikka. Eteenpäinen kinematiikka luokitellaan hieman kömpelöksi nykyään. Tekniikassa käytetään niin sanottua "top-down" systeemiä joka toimii siten, että esimerkiksi hahmon käden liikuttaminen vaatii ensin olkavarren kääntämistä, sitten kyynärvarren ja viimeiseksi käden liikuttamista halutulla tavalla. Animointi on yksinkertaista tällä tekniikalla ja toimii perusanimoinnissa. Haluamme

kuitenkin jotain paljon parempaa, tekniikkaa jossa hahmon liikkeet eivät ole kömpelöitä. Tämä vaatii tutustumista käänteiseen kinematiikkaan. (Pitzel 2017.)

Käänteinen kinematiikka käyttää hahmon luurankoa, joka on luotu rigaus vaiheessa. Luurangon rakenne on samantyyppinen kuin oma luurankosi, jokainen nivel pitää sen koossa. Käänteisessä kinematiikassa määritellään hahmolle ”root” eli juuri. Ihmishahmon juuri on usein selkärangan pohjassa. Luurangon käytön hyödyt tulevat siitä, kuinka voidaan liikuttaa esimerkiksi pelkkää kättä, ja käsivarren liikkeet tulevat perässä. (Pitzel 2017.)

5 Blender

Blender on ilmainen avoimen lähdekoodin 3D luomistyökalu. Blenderillä voidaan tehdä kaikki mallinnuksesta renderöintiin sekä videoiden muokkaamiseen ja pelien tekoon. Työkaluna Blender soveltuu yksityisille harrastajille sekä pienille yrityksille. (Blender About 2018.)

5.1 Historia

Blenderin tarina alkaa Ton Roosendaalista, joka halusi päivittää silloisen yrityksensä 3D-työkaluja vuonna 1995. Vuonna 1998, Ton perusti uuden yrityksen ja yritti kehittää ja markkinoida Blender-ohjelmaa lisää. Yritys ei kuitenkaan menestynyt kovin hyvin ja yrityksen investoinnit päättyivät vuonna 2002.

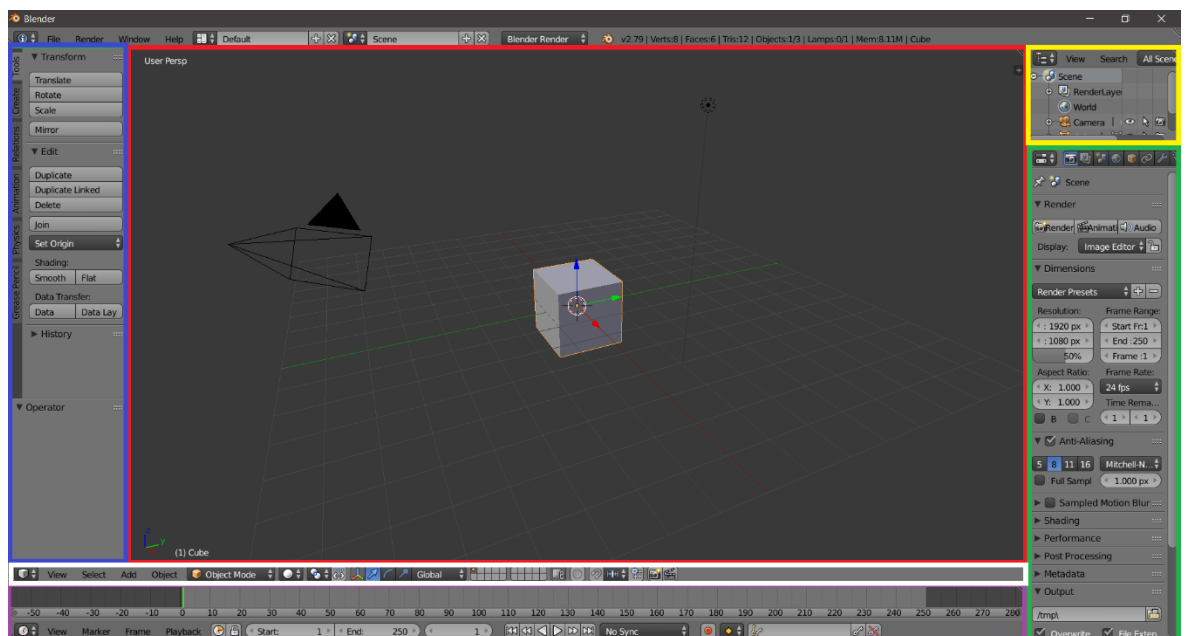
Blender oli kuitenkin jo saanut jonkin verran suosiota, eivätkä innokkaat käyttäjät suostuneet Blenderin katoamiseen. Ton Roosendaal päätti aloittaa yleishyödyllisen Blender Foundation -yrityksen vuonna 2002. Blenderin kehitystä jatkettiin avoimen lähdekoodin perusteella, vapaaehtoisella työvoimalla.

Blender yhteisö loi yhdessä avoimen elokuvan ”Elephants Dream”, jonka onnistuminen sai Ton Roosendaalin luomaan Blender Institute yrityksen vuonna 2007. Yrityksen tavoite on edistää Blender Foundation -yrityksen tavoitteita sekä koordinoita ja helpottaa avoimia 3D projekteja. (Blender Foundation 2013.)

5.2 Käyttö

Blenderin käyttö voi vaikuttaa vaikealta kun avaa ohjelman ensimmäistä kertaa, mutta käytön oppii nopeasti. Ensimmäisenä tulee vastaan käyttöliittymä. Kuvassa 1 nähdään Blenderin aloitusnäyttö. Väreillä reunustetut osiot ovat omia paneeleja, joilla on eri

toiminnallisuuksia. Punaisella reunustettu näyttö on 3D-näkymä, tällä näkymällä nähdään muokattavat objektit, sekä kaiken mitä 3D-maailmaan halutaan tuoda. Näkymä on suurin ja sen parissa tulee työskenneltyä eniten. Näkymän voi myös jakaa useisiin eri näyttöihin, ja valita minkä editorin haluaa näytölle. Keltaisella reunustettu osio on hierarkia näkymä, siitä näkee kaikki 3D-näkymässä olevat objektit. Sinisellä reunustettu osio on työkalut paneeli, tältä paneelilta löytää useita hyödyllisiä työkaluja jotka auttavat 3D objektin luonnissa. Paneelilta löytyy myös luonti välilehti, joka on yleensä 3D-projektin lähtökohta. Valkoisella reunustettu osio on erittäin tärkeä 3D näyttöön liittyvä paneeli. Tältä paneelilta löytyy sekä näytön ohjaamis painikkeet että objektin muokkaamis vaihtoehdot. Vihreällä reunustettu osio on projektin tarpeisto paneeli. Tällä paneelilla voidaan kustomoida objekteja enemmän, esimerkiksi materiaalien ja tekstuurien määrittely löytyy tarpeisto paneelistä. Näytön alla, lilalla reunustetulta osiolta, löytyy aikajana paneeli jota käytetään erityisesti animoinnissa.



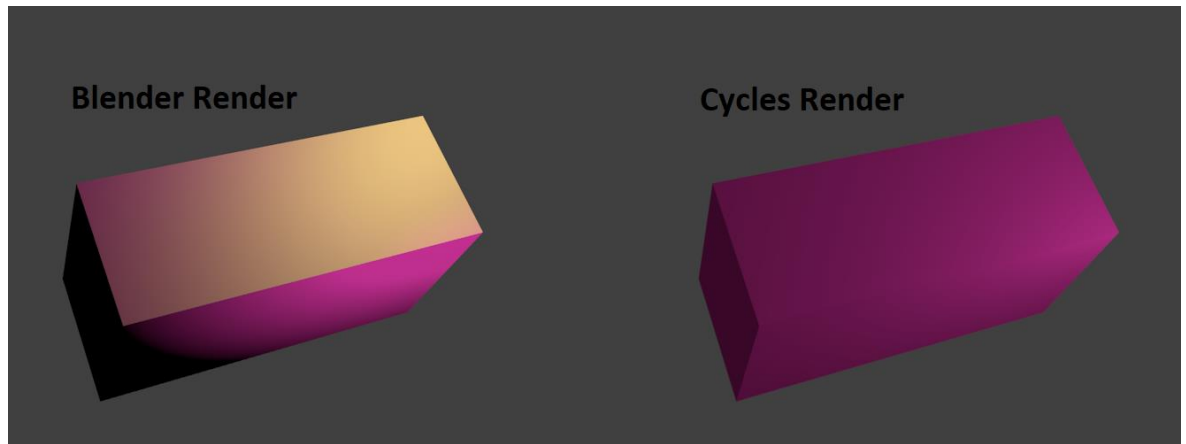
Kuva 1 Blender aloitusnäyttö

Navigointi 3D-näytöllä on yksinkertaisuudessaan hiiren painikkeet sekä sen liikkeet. Näytön kääntäminen tapahtuu painamalla hiiren keskipainike pohjaan, näytön liikuttaminen tapahtuu painamalla SHIFT-näppäin sekä hiiren keskipainike pohjaan. Näytön sisään ja ulospäin liikuttaminen hiiren keskipainikkeella skrollaamalla tai painamalla se pohjaan CTRL-näppäimen kanssa. Näiden ensimmäisten navigointi painikkeiden lisäksi Blender on täynnä eri näppäimistö oikoteitä ja niiden oppiminen voi viedä aikaa. Oikotiet ovat kuitenkin varsin hyvin dokumentoitu, ja niistä löytyy paljon ohjeita ja selityksiä. Objektien valinta tapahtuu painamalla hiiren oikeaa näppäintä, hiiren vasen näppäin määrittää 3D-kursorin paikan.

The image shows the top bar of the Blender 2.80 interface. The 'Object Mode' button is highlighted with a red rectangle, indicating the current active mode. Other visible buttons include 'View', 'Select', 'Add', and 'Global'.

Objekteille voi määritellä materiaalin oikean reunan tarpeisto valikosta. 3D-näytön näkymän voi muuttaa siten, että objektit näkyvät materiaali tilassa kuvan 2 vihreän osion painikkeella. Kaikkien objektien alkuperäinen materiaali on harmahtava väri, materiaali valikossa voidaan kuitenkin kustomoida objektin värin sekä muita materiaaliin liittyviä asetuksia.

Blender tukee useita renderöinti moottoreita, tärkeimpinä Blender Render ja Cycles Render. Blender Render on Blenderin vanhahko mutta nopea renderöinti moottori. Cycles Render on uudempi, hieman hitaampi moottori, mutta sen parempina puolina on sen realistinen renderöinti. Cycles tuottaa renderöityjä kuvia joissa objektit sekä valon lähteet käyttäytyvät realistisesti toisiinsa verrattuna (Kuva 3). (Surfaced Studio 2017.)



Kuva 3 Render moottoreiden vertailu

6 Unity

Unity Technologies tarjoaa grafiikkamoottorin jolla voidaan luoda 2D, 3D, ja virtuaalitodellisuus pelejä sekä sovelluksia monelle eri alustalle. (Unity About Us 2018.) Unity on ilmainen yksityisille käyttäjille sekä tarjoaa paketteja isoimmille yrityksille pelikehitykseen. Kaikilla Unity-pakettityypeillä voidaan luoda pelejä sekä myydä niitä, tosin ilmaisella pelimoottorilla voidaan tienata vain tietty määrä rahaa. (Unity Store 2018.)

6.1 Historia

Unity julkaistiin vuonna 2005, kaksi vuotta idean alullepanon jälkeen. Unity Technologies perustajat David Helgason, Nicholas Francis ja Joachim Ante saivat idean demokratisoida pelikehityksen ja mahdollistaa pelien kehityksen kaikille. Pelikehitys ja sen työkalut olivat kalliita ja useimmiten eivät soveltuneet yksityisten käyttäjien tai harrastajien käyttöön, Unity toi ratkaisun.

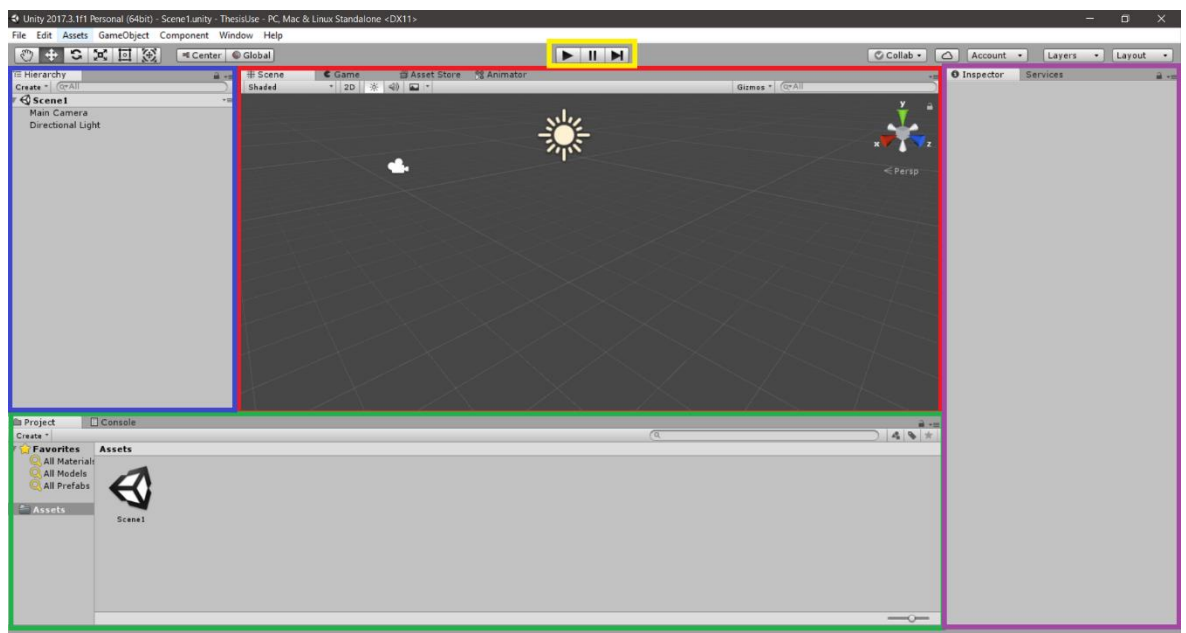
Unity on eniten käytetty pelimoottori markkinoilla, useat suuret ja pienet yritykset käyttävät Unityä luodakseen videopelejä sekä erilaisia sovelluksia. Unitylla voidaan luoda monipuolisempia pelejä ja sovelluksia useammille alustoille kuin muilla pelimoottoreilla. (Gamasutra 2010.)

6.2 Käyttö

Pelimoottorin käytön opettelu voi olla uhkaavaa, mutta Unity on yksi suosituimmista ja käyttäjäystävällisimmistä moottoreista. Unity tarjoaa paljon materiaaleja sen käytön taitamiseen sekä yksityiset käyttäjät ovat luoneet videoita ja artikkeleita eri aiheista liittyen Unityn käyttämiseen. Kun Unity avataan, tulee ensimmäiseksi näkyviin näyttö josta voidaan valita projekti jota halutaan työstää, tai luoda uusi 2D tai 3D projekti.

Avausnäytöllä on myös oppimis välilehti, josta löytyy paljon kursseja ja materiaaleja oppimiseen. Tässä projektissa on käytetty Unity 2017.3 versiota, mutta Unity on julkistanut version 2018.1.

Kuvassa 4 nähdään Unityn aloitusnäyttö. Punaisella rajattu alue on 3D-näyttö, sekä välilehdet pelinäkymään, ominaisuus kauppaan sekä animaattoriin. Uusia välilehtiä voi avata yläpaneelissa olevasta ikkuna valikosta. Keltaisella rajatut painikkeet aloittavat, keskeyttävät ja edistävät pelin kulkua, kun peli aloitetaan siirtyy 3D-näkymä pelinäkymään. Vihreällä rajattu alue on projektin ominaisuudet, kuvaan on luotu jo yksi skene. Ominaisuuksiin lisätään kaikki peliin vaadittavat tiedostot. Ominaisuus voidaan tuoda ulkoisesta ohjelmistosta tai luoda Unityssa. Sinisellä rajattu alue on projektin hierarkia johon aloituksessa lisätään kamera ja valo. Lilalla rajattuun alueeseen tulee projektin ominaisuuksien asetukset kun jokin objekti on valittuna.



Kuva 4 Unity aloitusnäyttö

Unitylla voidaan luoda monia erilaisia projekteja, mutta yleisimpiä ovat pelit. Pelit luodaan skene kerrallaan luomalla maailma näytölle ja pelillistämällä se. Peliobjektit ovat Unityn tärkeimpiä konsepteja, kaikki objektit pelissä ovat peliobjekteja. Peliobjekteille luodaan ominaisuuksia ja komponentteja. Komponentit ovat pelin objektien ja käytäntöjen a ja ö. Komponentteja voidaan lisätä ja muokata, sekä käyttää pelissä. Unityssa voidaan myös käyttää paljon ohjelmointia, koodilla voidaan tehdä monia asioita joita ei välittämättä voida tehdä suoraan käyttöliittymällä. (Unity User Manual 2018.)

Unitylla voidaan tehdä monia asioita. Ohjelman syvemmin opiskelu vie paljon aikaa, mutta pienempien osioiden oppiminen on helppoa. Unity on luonut erinomaiset dokumentaatiot kaikista ohjelmiston ominaisuuksista ja mahdollisuuksista. Lisäksi yritys sekä yksityiset henkilöt tarjoavat paljon apua omiin projekteihin. Riippuen siitä, mikä on projektin tavoite, löytyy usein erittäin paljon materiaalia jota kannattaa hyödyntää.

7 Hahmon mallinnus

Hahmon mallinnus toteutettiin hyvin yksinkertaisella tyyllillä. Tässä työssä haluttiin keskittyä kinematiikkaan ja hahmon liikuttamisen oppimiseen. Yksinkertainen hahmo helpotti rigausta sekä sen materialisointia. Hahmo ei ole realistinen, mutta se sopisi erinomaisesti esimerkiksi aloittelevan pelikehittäjän harjoituksiin.

7.1 Suunnittelu

Suunnittelu prosessi hahmon mallinnuksessa tapahtui hieman vahingossa. Hahmoa ei juurikaan alunperin suunniteltu, mutta suunnittelu tapahtui oppimisen ja kokeilun muodossa. Aluksi idea hahmolle oli aivan erilainen kuin lopputuloksen hahmo, mutta yrityksen ja erehdyksen kautta hahmo luotiin. Yrityksen ja erehdyksen kautta oppiminen lisääntyy, ja lopulta isot ja pienet virheet jäävät pois. Suunnittelu vaiheessa on hyvä ottaa huomioon omat taidot ja mahdollinen aikaraja. Jos suunnitelma on liian vaikea, on helppo lannistua ja lopputulos ei ole ollenkaan niin hyvä kuin omia taitoja vastaava tulos. Tämä saattaa myös myöhästyttää projektia huomattavasti. Itse hahmon suunnittelussa tulee huomioda mihin asentoon hahmo mallinnetaan. Hahmon asentoa tullaan muuttamaan myöhemmin animaatiota varten, mutta rigaus vaikeutuu, mikäli hahmo on erikoisessa asennossa. Tämän vuoksi hahmo usein mallinnetan T-asentoon, jossa hahmo seisoo suorassa, kädet sivuille ojennettuna. Asento myös auttaa hahmon viemistä muihin ohjelmistoihin.

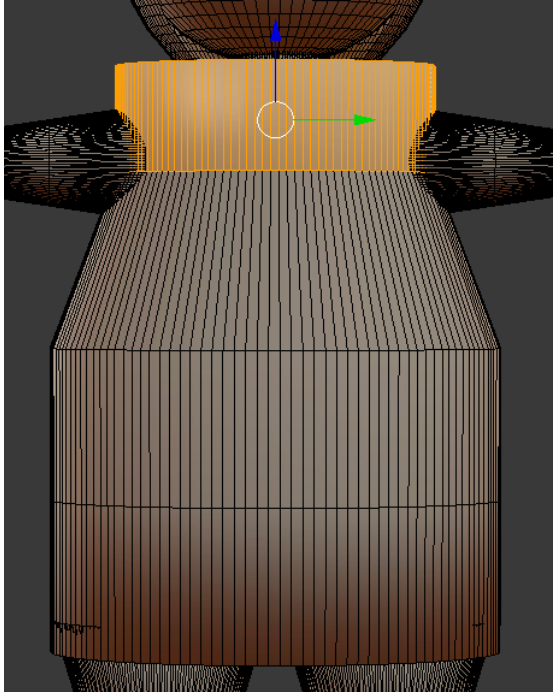
7.2 Toteutus

Hahmon toteutus aloitettiin luomalla uusi projekti Blenderillä ja tyhjämallä näyttö. Blender on hyvin käyttäjäystävällinen ohjelmisto, kunhan jaksaa opetella ja harrastaa sen käyttöä. Ensimmäistä projektia tehdessä tulee käytettyä vain murto-osaa kaikista Blenderin ominaisuuksista, ja jopa vuosien harrastamisella, ei kaikki ominaisuudet ole välittämättä tulleet eteen. Hahmon luomisessa käytettiin omia aiemmin opittuja taitoja. Ja vaikka hahmo onkin hyvin yksinkertainen, tuli muutamia osia toistaa ja parannella.

7.2.1 Hahmon luominen

Ensimmäisessä askeleessa luotiin hahmolle jalat. Hahmon jaloiksi käytettiin valmista "Cone" mallia, joka löytyy Blenderistä "Create" sivupalkin alta. Malli käännettiin siten että terävä pää osoittaa alaspäin, sekä mallin kokoa muutettiin ohuemmaksi. Malli kopioitiin ja siirrettiin vierelle emuloimaan toista jalkaa. Seuraava askel oli luoda hahmon torso. Torsoon käytettiin valmista "Cylinder" mallia. Tämän kokoa muutettiin siten, että se sopi

jalkojen kokoon. Lisäksi torson yläosaa muutettiin pienemmäksi kuin alaosa. Tämä saavutettiin valitsemalla halutut kasvot mallista muokkaus tilassa ja pienentämällä osiota (Kuva 5). Tällä tavalla yläosa on suora, mutta sen alla olevat kasvot ovat kaltevat ja alaosa pysyy suorana.



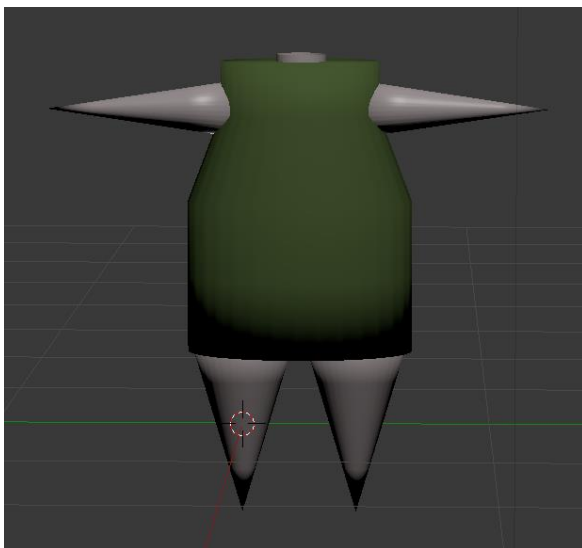
Kuva 5 Torson muokkaaminen

Tyytyväisenä torsoon, voitiin jatkaa käsien luomiseen. Kädet luotiin samalla tyylillä kuin jalat. Käytettiin valmista mallia "Cone", ja käännettiin se kallelleen sekä muokattiin sen kokoa. Tämän jälkeen käsi kopioitiin, käännettiin se 180 astetta ja siirrettiin oikealle kohdalle (Kuva 6).



Kuva 6 Torso, kädet ja jalat

Kädet ja jalat voidaan yhdistää torsoon valitsemalla kaikki objektit käyttämällä "Join" toimintoa "Tools" sivupalkin alta. Yhdistämisen jälkeen torsoa ja raajoja voidaan muokata helposti samanaikaisesti eikä raajat irtoa torsosta kun mallia liikutetaan. Seuraavaksi luotiin hahmolle kaula, tähän käytettiin valmismallia "Cylinder", sen kokoa muutettiin ja se siirrettiin oikealle kohdalle, sekä yhdistettiin torsoon (Kuva 7).



Kuva 7 Hahmo ilman päätä

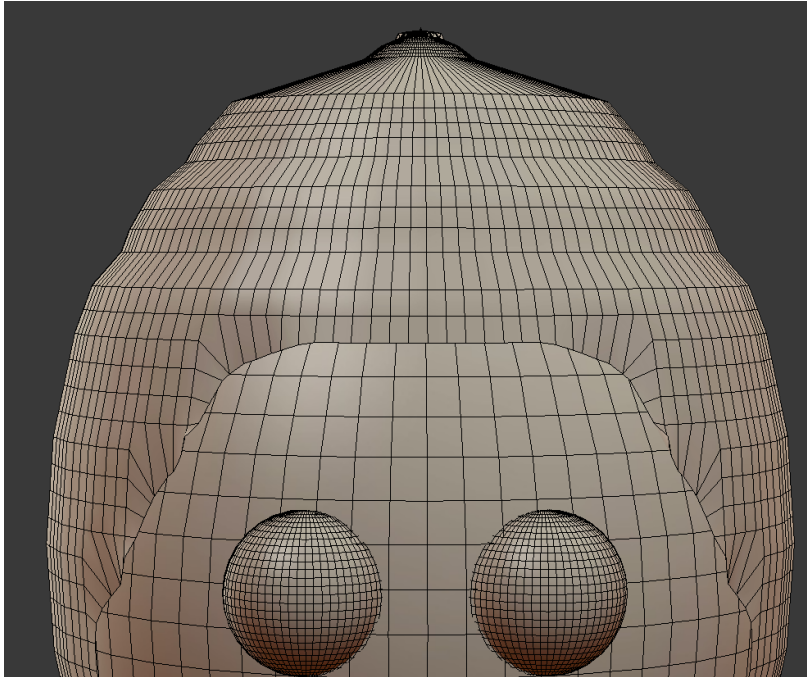
Seuraava vaihe on luoda hahmolle pää. Pään pohjana käytettiin valmismallia "UV Sphere", joka on pallon muotoinen. Malli muutettiin muodostamaan ovaalin, tämä tapahtuu helposti pienentämällä mallia X- ja Y-akseleilla. Tyytyväisiä pään muotoon ja kokoon, siirretään se oikeaan kohtaan, kaulan päälle. Yksinkertaisuuden nimissä luotiin hahmolle silmät luomalla uusi pieni "UV Sphere" ja siirtämällä se oikeaan kohtaan. Silmä

kopioitiin ja siirrettiin toisen vierelle. Hahmon suu luotiin käyttämällä valmismallia "Torus". Suu pienennettiin ja siirrettiin kutakuinkin oikealle kohdalle. Suuta täytyy kääntää Y-akselilla jonkin verran, asteiden määrä riippuu pään kohdasta johon suu asetetaan. Kun suu on tarpeeksi kallellaan, voidaan suu siirtää oikealle kohdalle (Kuva 8).



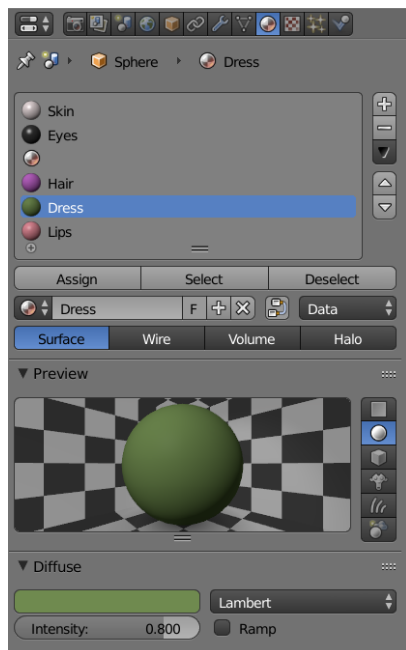
Kuva 8 Silmät ja suu

Viimeiseksi luotiin hahmolle huppu lisäämään siihen vähän jotain erikoista. Huppu tehtiin muokkaamalla uutta "UV Sphere" muotoa ja sen kokoa muutettiin samalla tavalla kuin pään kokoa, tehden siitä ovaalin muotoisen. Siirrettäessä hupun kohdalle, johon se tullaan asettamaan, nähdään mitkä kohdat siitä tulisi poistaa, jotta hahmon kasvot näkyvät läpi. Muokkaustilassa voidaan poistaa objektin yksittäisiä kasvoja valitsemalla ne ja poistamalla. Aukon luomisen lisäksi pienennetään hupun yläosaa useasta kohdasta. Hupun aukon reunoja siirrettiin muokkaus tilassa, jotta se ei olisi teräväkulmainen (Kuva 9).



Kuva 9 Hupun yksityiskohdat

Hahmon mallinnus on lähes valmis. Ennenkuin yhdistetään kaikki objektit toisiinsa, lisätään niille materiaaleja. Hahmolle luotiin yksinkertaiset väri materiaalit mekolle, jota myös huppu seuraa sekä silmille, huulille ja iholle (Kuva 10).



Kuva 10 Materiaalit

Kun objekteja ei ole vielä yhdistetty, on helppoa lisätä niille eri materiaalit. Valitsemalla objektin kuten silmän, voidaan muokkaus tilassa asettaa sille oma materiaali käyttämällä ”Assign” toimintoa. Jos objekteja on yhdistetty toisiinsa, tai halutaan eri materiaali jollekin

yksittäiselle kohdalle, voidaan muokkaustilassa valita ne kasvot joille halutaan eri materiaali, ja sitten asettaa se vain näille kasvoille.



Kuva 11 Valmis hahmon malli

7.2.2 Rigaus

Luurangon luomisessa on otettava huomioon mihin tarkoitukseen hahmoa käytetään. Tässä työssä hahmo tullaan viemään Unity pelimoottoriin, ja pelimoottorilla on omat säännöt siihen mitkä luut se vaatii. Työn teko hetkellä Unity vaatii hahmolle seuraavat luut: lonkka, selkäranka, pää, yläkäsi, alakäsi, käsi, yläjalka, alajalka, ja jalkaterä. Muitakin luuita voidaan lisätä, mutta ne eivät ole pakollisia hahmon toiminnallisuuteen Unityssä. Lisäksi on hyvä, jos hahmo on mallinnettu T-asennossa, tämä helpottaa sen rigausta, sekä sen viemistä Unityyn.

Hahmon rigauksen ja kinematiikan luonnin apuna käytettiin Pluralsight artikkelia ”How to Create Your First Character Rig in Blender: Part 1 - Setting up the Armature” (Pluralsight 2015.) sekä Sebastian Laguen YouTube videota ”Blender Tutorial: Basics of Character Rigging” (Sebastian Lague 2013.).

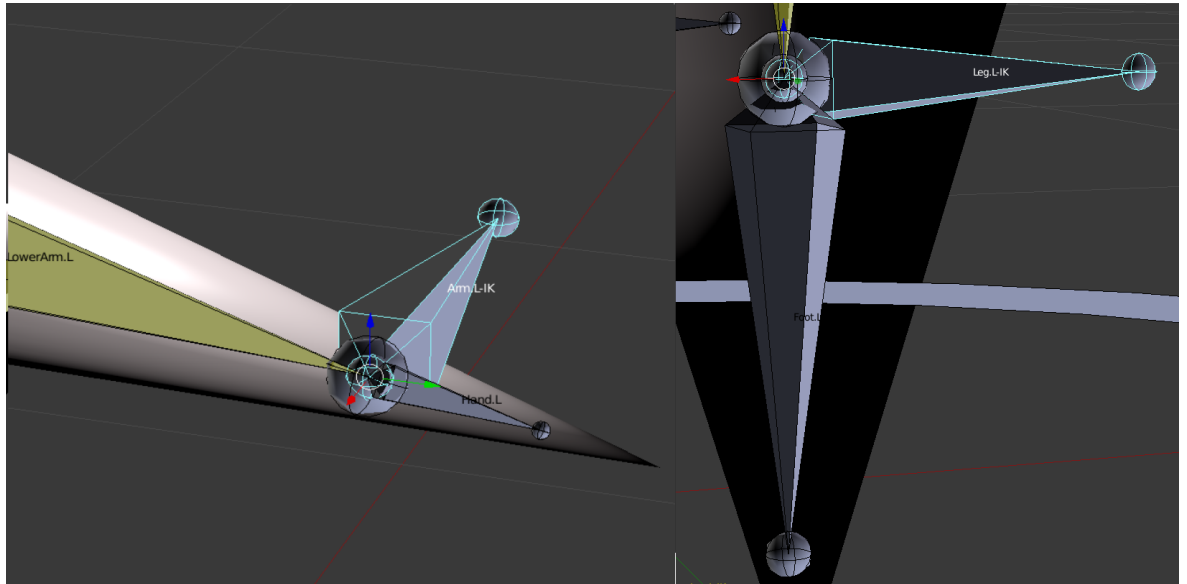
Hahmon rigaus aloitetaan luomalla ensimmäinen luu valitsemalla ”Add” valikosta ”Armature – Single Bone”. Tällä tavoin luodaan yksi luu. Luun kokoa ja orientaatiota voidaan muokata tarpeen tullen, ja sitten siirtää se oikealle kohdalle. Hyvä aloituskohta luurangolle on lonkaluu, se voidaan asettaa hahmon vartalon alaosaan keskelle, jakojen

ylle. Lonkaluuhun yhtyviä luita voidaan lisätä "Extrude" toiminnolla ja vetämällä ne oikeisiin kohtiin. Lonkaluusta ylöspäin luodaan selkärangan luu, ylävartalon luu, kaulan luu sekä pään luu (Kuva 12).



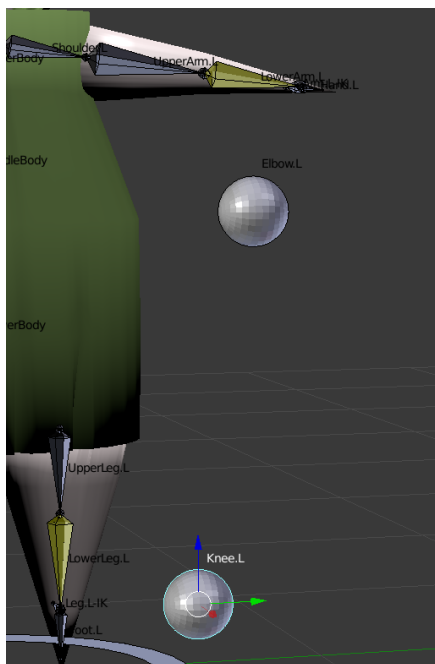
Kuva 12 Vartalon rigaus

Seuraavaksi luodaan olkapään luu lisäämällä uusi erillinen luu. Olkapään luusta eteenpäin kämmentä kohti voidaan luoda yläkäden-, alakäden-, ja kämmenen luut. Jalan luut luodaan samalla tavalla, aloittamalla yläjalasta, päätyen jalkaterään. Ennen kun kopioidaan käden sekä jalan luut hahmon toiselle puolelle, luodaan kämmenelle ja jalkaterälle IK (Inverse Kinematics) luut. Nämä luut asetetaan alakäden ja kämmenen väliin, sekä alajalan ja jalkaterän väliin (Kuva 13). IK-luiden asetuksissa on tärkeää poistaa asetus "Deform", jotta nämä luut eivät vaikuta hahmon vartalon muotoihin.



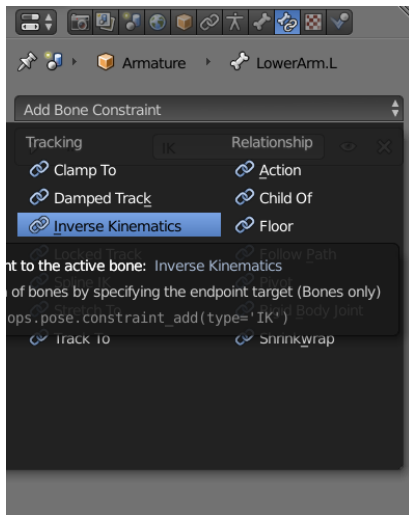
Kuva 13 IK luut

Luut tarvitsevat myös ohjenuoran sille, mihin suuntaan käsi ja jalka voivat taittua, ja miten ne taittuvat. Tähän tarpeeseen lisätään kaksi luuta, kyynärpää luu ja polvi luu. Nämä luodaan kuten muutkin luut, mutta asetetaan erilleen hahmosta (Kuva 14).



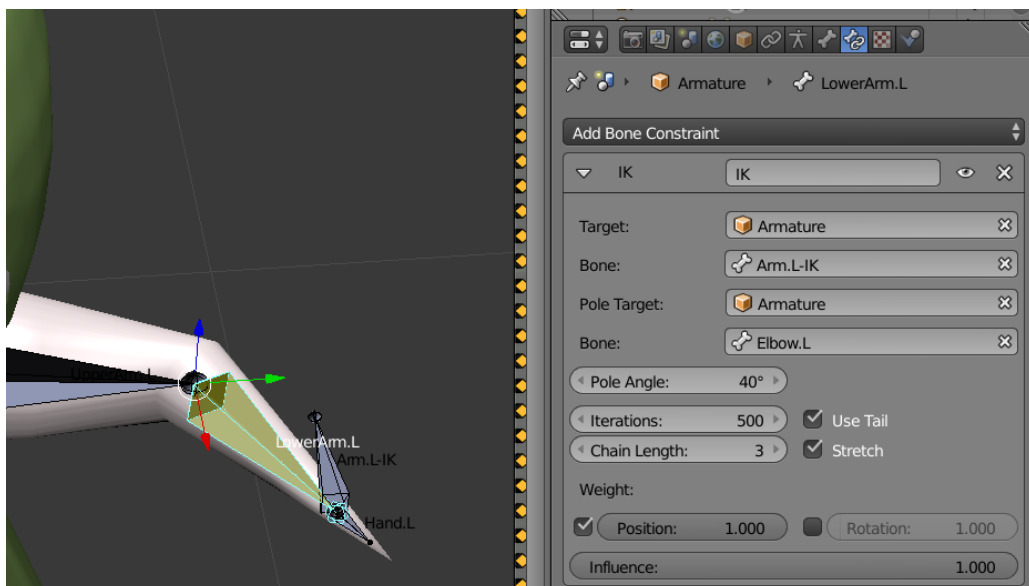
Kuva 14 Kyynärpää ja polvi luut

Seuraavaksi asetetaan kinematiikka. Kinematiikan asetuksessa valitaan alakäden luu, oikeassa reunassa olevassa valikossa voidaan luoda uusi luu rajoite (Kuva 15).



Kuva 15 Kinematiikan asetus

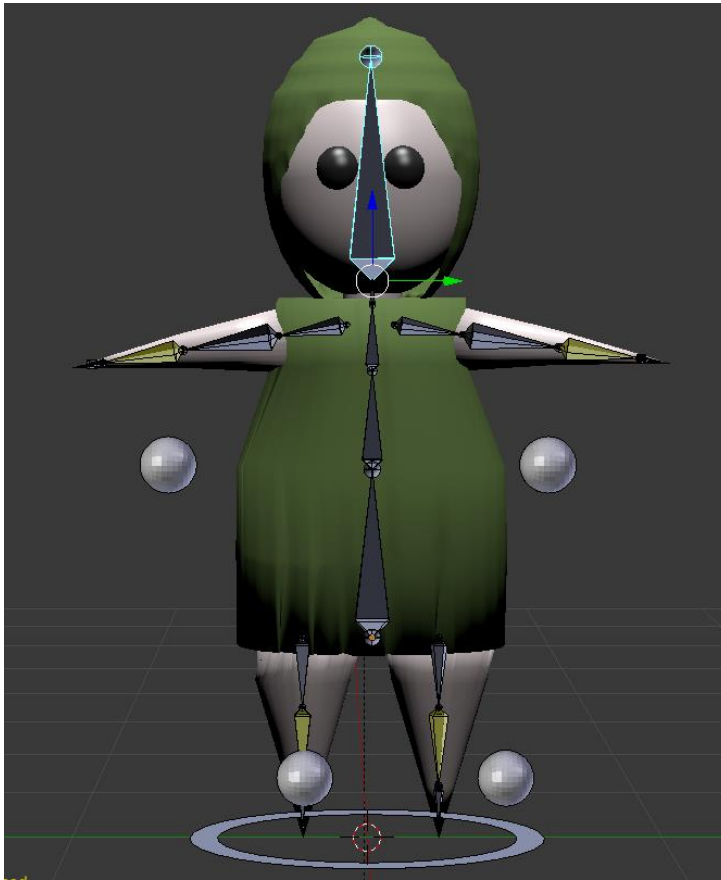
Kinematiikan asetuksessa määritellään se, mikä luu ohjaa tätä kyseistä luuta, joka tässä tapauksessa on käden IK-luu. Tämän lisäksi voidaan asettaa kyynärpään luu. Tärkeää on muistaa lisätä myös luun ketjun pituus, joka tässä tapauksessa on kolme, tämä sisältää olkapään luun, yläkäden luun ja alakäden luun (Kuva 16). Näiden lisäksi tulee usein muokata missä kulmassa kyynärpään luu taittaa kättä, asetus vaatii kokeilua siitä mitkä kulmat sopivat kyseisille luille. Jalan kinematiikka luodaan samalla tavalla, sen luiden ketjun pituus on kuitenkin tässä mallissa vain kaksi.



Kuva 16 Käden kinematiikka

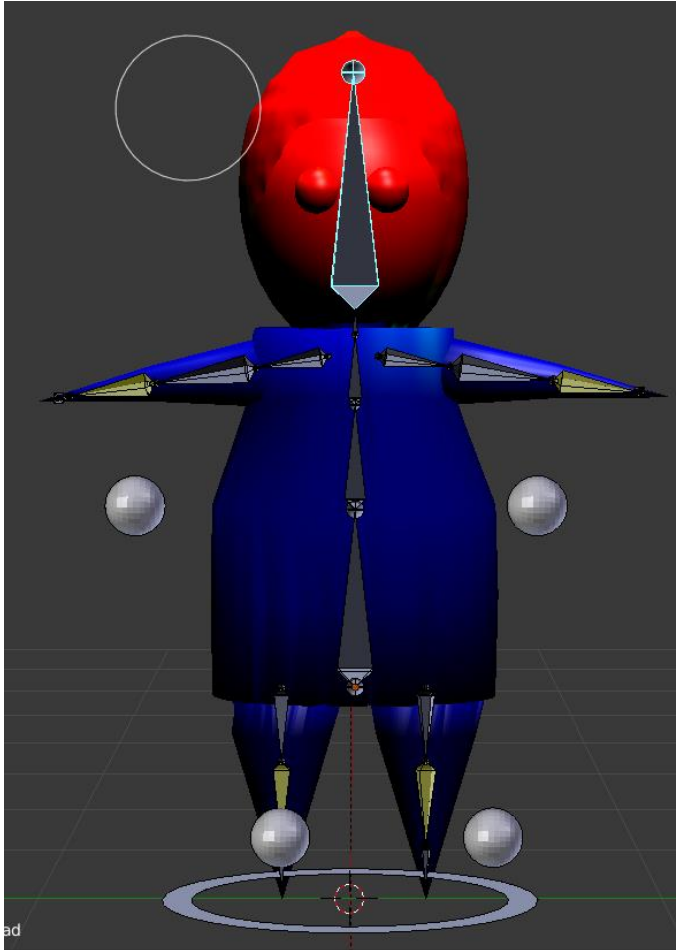
Kun kinematiikka on asetettu, voidaan kopioida luut myös hahmon toiselle puolelle. Valitsemalla kaikki käden ja jalan luut, sekä IK-luut, kyynärpään ja polven luut, kopioidaan luut ja ne voidaan kääntää "Armature" valikosta "Mirror" toiminnolla. Luiden asetetut nimet voidaan muokata joko käsin, tai "Armature" valikossa "Flip Names" toiminnolla. Viimeinen

luu joka tulee lisätä on kontrolliluu. Kontrolliluu asetetaan erillisenä hahmon jalkojen väliin (Kuva 17).



Kuva 17 Valmiit luut

Luut tulee nyt yhdistää hahmo objektiin. Valitaan luut ja hahmo, "Object" valikosta "Parent" ja toiminto "With Automatic Weights". Tämä toiminto yhdistää luut hahmoon ja luo niille automaattisesti painot. Automaattisesti asetetut painot tulee tarkistaa "Weight Paint" näkymässä. Kuvassa 18 nähdään pään paino, punaisella olevat pinnat ovat täysin pään luun hallinnassa, eikä luu hallitse ollenkaan sinisiä pintoja. Näkymässä voidaan myös lisätä ja poistaa painoa eri luilta, jolloin hahmon liikuteltavuus toimii parhaiten.



Kuva 18 Weight Paint näkymä

Kinematiikka on lähes täysin valmis. Viimeisenä tulee luoda luiden väleille yhteyksiä. Olkapään luut tulee yhdistää selkärankaan, yläjalkojen luut tulee yhdistää lonkkaluuhun. Valitsemalla halutut luut voidaan niiden välille muodostaa yhteys valikosta "Armature – Parent". Tämän lisäksi, tulee IK-luiden, polvi- ja kyynärpää luiden sekä lonkkaluun tulee muodostaa yhteys kontrolliluuhun (Kuva 19). Kontrolliluulla voidaan siten liikuttaa koko hahmoa.



Kuva 19 Luiden väliset suhteet

Hahmon luuranko ja kinematiikka on näillä askeleilla valmis. Kinematiikan avulla hahmoa voidaan liikuttaa eri asentoihin (Kuva 20).

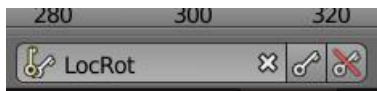


Kuva 20 Hahmon eri asento

7.2.3 Animointi

Hahmolla tulee olla animoituja asentoja, jotta voimme luoda kävely syklin Unityllä. Kun kinematiikka on luotu hahmolle hyvin, on erittäin helppo animoida hahmo Blenderillä. Tämän työn animoinnissa käytettiin apuna Sebastian Laguen YouTube videota "Blender Tutorial: Basic Walk Cycle" (Sebastian Lague 2012.).

Hahmon animointi aloitetaan jakamalla Blender näyttö kahteen, toisella näytöllä tulee olla hahmo "Pose" tilassa, uudelle näytölle valitaan "Dope Sheet" tyyppi. Animaation "Active Keying Set" arvoksi tulee muuttaa "LocRot" (Kuva 21). "LocRot" tallentaa jokaisen luun sijainnin ja käännön.

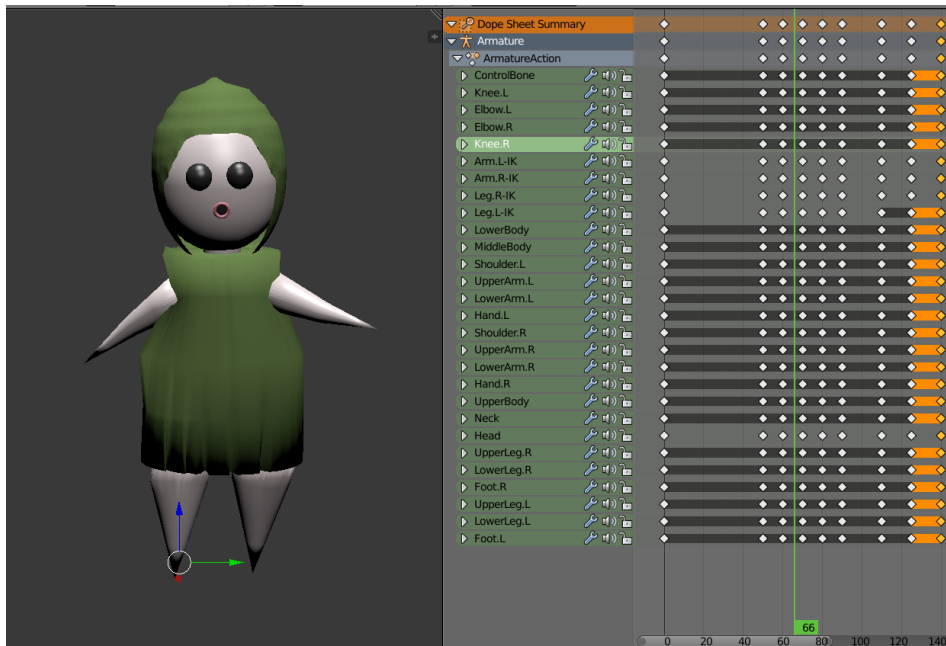


Kuva 21 Active Keying Set

Aloittaessa on hyvä tallentaa T-asento ensimmäiseksi animoinnin raamiksi. Animointi raamin luonnissa tulee ensin valita kaikki luut painamalla "A"-näppäintä, ja sitten painaa näppäintä "I", tämä lisää raamin "Dope Sheet":iin. "Dope Sheet" näkymässä voidaan siirtää palkkia joka määrittää mitä raamia sillä hetkellä käsitellään. Kävely syklin luomiseksi tulee palkkia siirtää eteenpäin. Kävely syklin aloittamiseksi voidaan liikuttaa hahmon jalkoja sekä käsiä emuloimaan ensimmäistä askelta. Kun raajat ovat liikutettu haluttuihin asetuksiin, valitaan kaikki luut ja lisätään asento "Dope Sheet":iin. Animoinnin raami on nyt liikutettava eteenpäin, raamin numero riippuu siitä kuinka nopean liikkeen haluaa. Suurempi väli raameissa tekee askeleesta hitaan kun taas lyhyt väli tekee liikkeistä nopeita. Uudessa raamissa tulee asettaa jalat ja kädet uudestaan haluttuihin asentoihin jonka jälkeen se voidaan lisätä "Dope Sheet":iin. Kaikki askeleen vaiheet luodaan edellä mainitulla tavalla, kun päästään syklin loppuun, voidaan kopioida ensimmäinen askel raami ja asettaa se viimeiseksi syklin raamiksi. Tällöin animaatiota voidaan käyttää vähentämällä aivan viimeinen raami animoinnin kulusta, sykli näyttää saumattomalta (Kuva 22).

Hahmolle voidaan lisätä useita animaatioita, joita voidaan siten käyttää Unityssä, mutta tässä työssä on tarve ainoastaan kävely syklille sekä joutokäynti syklille. Joutokäynti eli "Idle" sykli on se mitä hahmo tekee kun se seisoo paikallaan. Liikkeet voivat olla erittäin pieniä tai isompia huiskauksia. Syklin toteutus tapahtuu samoin kuin kävely syklin

toteutus, on kuitenkin muistettava jättää jonkin verran raameja syklien väliin. Tallennettaessa Blender projektin, animaatio tallentuu sen mukana.



Kuva 22 Kävely syklin animaatio

8 Hahmon liikkuttaminen pelimoottori Unitylla

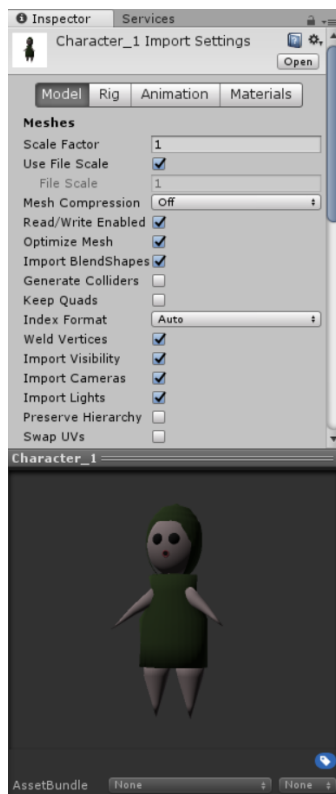
Tämän työn Unity osiossa käytettiin apuna Darrin Lilen YouTube videoita "Blender Character To Unity part 1 of 2" ja "Blender Character To Unity part 2 of 2" (Darrin Lile 2015.). Unity myös tarjoaa paljon dokumentaatiota ja oppimateriaaleja kaikenlaisten projektien luomiseen. Työtä tehdessä tekijän taidot Unityn käytössä olivat erittäin alkeelliset. Työn edetessä taidot kuitenkin lisääntyivät, ja nyt saman toteutuksen luominen olisi paljon helpompaa.

8.1 Hahmon importointi Unityyn

Hahmon mallinnuksen jälkeen on aika viedä hahmo pelimoottoriin. Tallennetun Blender tiedoston voi helposti importoida Unityyn. Avatessa Unity, tulee luoda uusi 3D projekti. Uudessa projektissa tallennetaan uusi skene, tämä kannattaa tehdä jo ennen hahmon importointia. Seuraava vaihe on tuoda hahmo Unityyn. Unity importtaa Blender tiedostot käyttäen Blenderin FBX-vientiä, joka tukee rigattuja ja animoituja objektien vientiä (Blender 2.79 Manual). Hahmon voi helposti importoida raahaamalla Blender tiedosto tietokoneen kansiota avoinna olevan Unity projektin "Assets" kohtaan. Vaihtoehtoisesti Blender tiedosto voidaan suoraan tallentaa projektin "Assets" kansioon, jolloin hahmo tulee näkyviin projektiin. Blender tiedoston tallentaminen suoraan projektiin on hyödyllistä,

sillä muuttaessa Blender tiedostoa, sen tallentuessa tiedosto päivittyy myös Unity projektissa. (Unity Documentation 2017.)

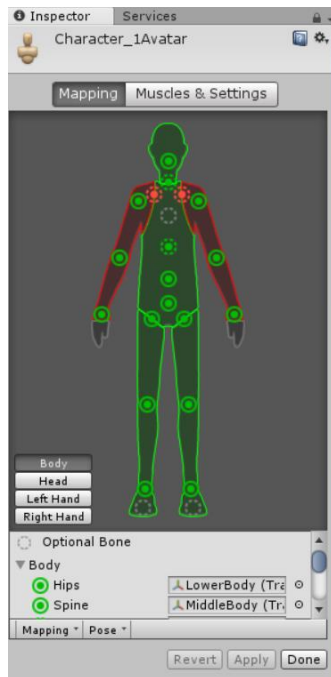
Kun hahmo on importattu Unityyn, voidaan sitä klikata ja tarkastella ”Inspector” sivupalkista sen tuontitietoja. Tuontitiedoissa ensimmäiseksi näkyvässä välilehdessä ”Model” eli ”Malli”, voidaan tarkastella hahmon importoimisen perustietoja, tässä projektissa voidaan pitää alkuperäiset tiedot, eikä niitä tarvitse muuttaa (Kuva 23).



Kuva 23 Unity Inspector

Seuraavassa välilehdessä ”Rig”, tulee importoida hahmon rigaus, eli noin sanottu luuranko. Animaation tyyppiä valitaan ”Humanoid”. Tämän tyyppinen animaatio tarkoittaa että hahmon rig on ihmistyyppinen, eli sillä on kaksi jalkaa, kaksi kättä ja pää. Unity havaitsee luurangon automaattisesti ja osaa asettaa luut oikein myös Unity projektissa. Joissain tapauksissa saattaa joutua määrittelemään tai korjaamaan luita konfigurointi kohdasta. (Unity Documentation 2017.) Animaatio tyyppiä valittua tulee painaa ”Apply” painiketta, joka alkaa Unity prosessin, jossa Unity pyrkii automaattisesti määrittelemään luut ja niiden kohteet. Jos Unity saa luurangon määriteltä oikein, näkyy ”Configure...” painikkeen vierellä oikein-merkki. Jos luurankoa ei ole määriteltä täysin oikein, tulee se konfiguroida manuaalisesti (Kuva 24). Konfiguroinnissa voidaan määrittellä luut manuaalisesti. Konfigurointi tulee tarkistaa vaikka oikein-merkki tulisikin näkyviin, sillä on mahdollista että Unity on määritellyt jotkut luut väärin ja ne tulee korjata manuaalisesti.

Konfiguroinnin lisäämisen jälkeen, tulee kokeilla uudestaan saada oikein-merkki "Rig" sivulle.

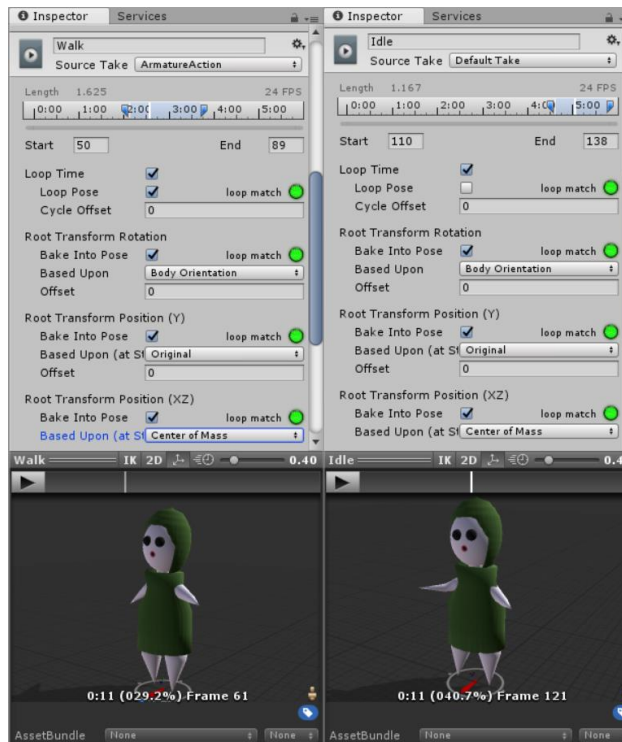


Kuva 24 Luurangon konfigurointi

Rigauksen importoinnin jälkeen siirrytään animoinnin välilehteen. Animoinnin lisäämisessä tulee muistaa raamit, joihin on asetettu animointia Blenderissä. Suurimmaksi osaksi asetukset voidaan jättää alkuperäisiin asetuksiin Unityssä, animoidut liikkeet ovat kuitenkin asetettava itse. "Clips" näkymässä nähdään Unityn itse luomat animaatio syklit, ne eivät kuitenkaan ole oikein. Unityn luomat animaatiot voidaan poistaa ja lisätä uusi animaatio painamalla "plussa"-merkkiä. Animaatiolle on annettava nimi ja määriteltävä sen lähde. Kävely-animaation lähteeksi asetetaan "ArmatureAction", tyhjäkäynti-animaation lähteeksi asetetaan "Default Take". Seuraavaksi on määriteltävä animaation aikalinja. Kävelysyklin aloittava raamin numero asetetaan animaation aloituskenttään, loppukenttään tulee asettaa kävelysyklin viimeistä raamia edeltävän raamin numero. Kävelysykli voi esimerkiksi alkaa raamista 50 ja loppua raamiin 89, vaikka animaation viimeinen raami olisikin 90. Viimeisen raamin pois jättäminen takaa sen, että kävely todella on syklimäinen ja se jatkuva. Raamiasetuksen jälkeisien asetusten perässä olevat "loop match" ympyrät, tulisi kaikkien näyttää vihreätä valoa. Täten animaationsyklin tulisi toimia hyvin ja Unity tunnistaa animaation.

Seuraavista asetuksista on tärkeää asettaa käyttöön "Loop Time"-asetus, joka aloittaa animaation uudestaan kun viimeinen raami on ajettu. "Loop Pose"-asetus otetaan myös käyttöön, tämä takaa animaation saumattomuuden. Seuraavat eri akselien "Root

Transform Rotation"-asetukset voidaan jättää kutakuinkin alkuperäisiin asetuksiin, valitaan kuitenkin käyttöön kaikki "Bake Into Position"-asetukset. Asetukset takaavat hahmon pysymisen oikeassa asennossa kävely syklin aikana. Muut asetukset voidaan jättää siten, kuin Unity on ne määritellyt. Animaation toimivuutta voidaan testata asetuksien alta, jossa hahmo toistaa asetetut animaatiot. Tyhjäkäynti animaation asetus käy täysin samalla tavalla, animaation lähteeksi tosin asetetaan "Default Take" (Kuva 25).



Kuva 25 Animaatiosyklin asetukset

Viimeinen importoinnin asetus on materiaalit. Hahmon materiaalit asetettiin kuitenkin suoraan objektiin, joten materiaaleja ei ole tarpeellista asettaa Unityssä.

8.2 Animaation asettaminen hahmolle

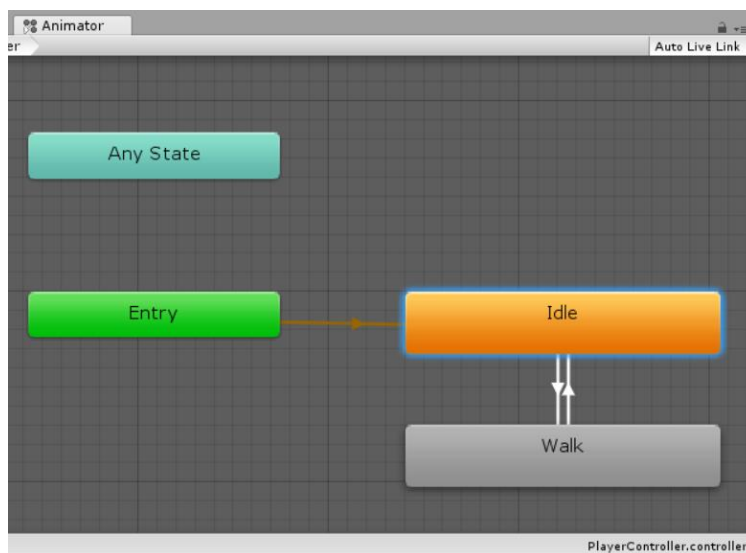
Animaation importointi ei vielä aseta animaatiota hahmolle, hahmo ei siis vielä liiku tai aja tyhjäkäynti sykliä kun hahmo on raahattu pelinäkymään. Hahmolle on asetettava animaatio kontrolleri, joka määrää mitä animaatioita hahmo käyttää.

Animaatiokontrolleri lisätään hahmolle lisäämällä projektin "Assets" kohtaan uusi Animaattori kontrolleri. Kontrolleri voidaan suoraan asettaa hahmon animaatiokontrollerin kohdalle. Ikkunat-valikosta tulee valita "Animator", jolla voidaan määrittää animaatiokontrollerille tehtäviä. Hahmon ominaisuuksissa näkyy luodut animaatiot, tässä projektissa näkyvissä tulee olla kävely ja tyhjäkäynti animaatiot. Ensimmäiseksi hahmolle

asetetaan tyhjäkäynti animaatio, tämä tapahtuu helposti raahaamalla tyhjäkäynti animaation tiedosto hahmon ominaisuuksista animaattoriin. Animaattori asettaa animaation oikeaan kohtaan, ja nyt kun ajamme pelin, hahmo liikkuu tyhjäkäynti animaation mukaan.

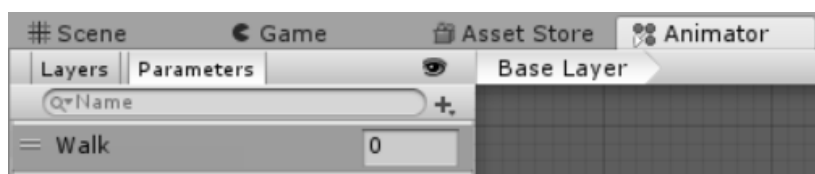
Kävely animaation lisääminen tapahtuu kutakuinkin samalla tavalla. Kävely animaatio voidaan raahata animaattoriin ja asettaa se tyhjäkäynti animaation lähelle.

Animaatiosykli vaativat yhteyksen toisiinsa. Klikkaamalla hiiren oikean puolista painiketta tyhjäkäynti animaation päällä voimme valita valikosta "Make transition"-valinnan. Valinta tuo näytölle nuolen jonka lähtökohta on tyhjäkäynti animaatiossa. Raahaamalla nuolenpään kävely animaatioon, teemme niiden välille yhteyden. Toinen yhteys vaaditaan, jotta kävely animaatiosta voidaan siirtyä takaisin tyhjäkäyntiin. Yhteys luodaan samoin, mutta yhteyden lähtökohta on kävely animaatio (Kuva 26).



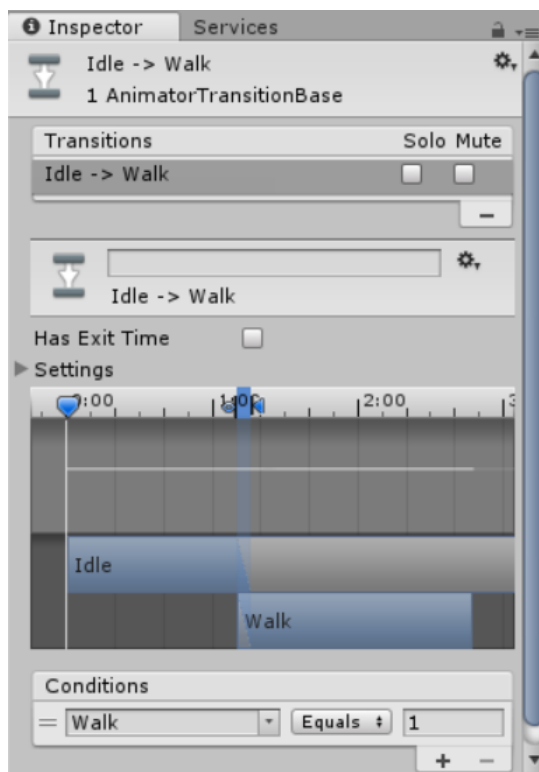
Kuva 26 Animaattori

Seuraavaksi tarvitaan tapa jolla voidaan siirtyä animaatiosta toiseen. Animaattori näkymässä voidaan valita välilehti "Parameters". Tällä välilehdellä voidaan luoda parametrejä, jotka määrittävät koska siirrytään animaatiosta toiseen. Plussa-merkkiä painamalla luodaan uusi parametri jonka tyyppi on kokonaisluku eli "Int". Parametri nimetään tässä työssä "Walk"-nimiseksi ja sen arvoksi jätetään nolla (Kuva 27).



Kuva 27 Parametri-välilehti

Parametri tulee vielä asettaa animaatioyhteyksiin. Painamalla nuolta tyhjäkäynti ja kävely animaatioiden välillä, saamme näkyviin muutosasetuksia. Muutosasetuksissa voidaan asettaa edellytyksiä, jotka määrittelevät sen minkä arvoinen "Walk"-parametrin tulee olla, jotta animaatiomuutos tapahtuu. Lisätään uusi edellytys, parametrin nimi näkyy automaattisesti, alavetolaatikosta valitaan kuitenkin "Equals" eli on yhtäsuuri kuin. Riippuen kumpaa muutosta ollaan muokkaamasta viimeinen arvo on joko nolla tai yksi. Kun siirrytään tyhjäkäynti animaatiosta kävely animaatioon, tulee arvon olla yksi (Kuva 28). Kävely animaatiosta siirtyminen tyhjäkäyntiin, arvo jätetään nolaksi. Molemmista muutoksista kannattaa myös ottaa valinta "Has Exit Time" pois. Tämän valinnan poistaminen takaa sen, että animaatio muuttuu ilman viivästyksiä ja muutos on nopea. Muitakin asetuksia saattaa olla tarpeen muuttaa, mutta tarvittavat muutokset on helpompi havaita vasta kun hahmon liikuteltavuus on muutoin valmis.



Kuva 28 Tyhjäkäynnistä kävelyyn muutos

8.3 Hahmon pelillistäminen

Jotta voidaan käyttää edellä määriteltyjä asetuksia, tulee luoda uusi ohjelmakoodi.

"Assets" osiossa klikkaamalla hiiren oikeaa näppäintä ja valitsemalla luo, luodaan uusi C# ohjelmakoodi. Nimetään luotu ohjelmakoodi, tässä projektissa käytetään nimeä "Player". Seuraavaksi avaamme ohjelmakoodin ja aloitamme sen kirjoittamisen.

Kaksoisklikkaamalla ohjelmakoodia, Unity yleensä avaa oman MonoDevelop nimisen koodieditorin mutta on myös mahdollista valita jokin toinen editori.

Ohjelmakoodi on lähestulkoon tyhjä kun se avataan ensimmäistä kertaa. Koodissa nähdään funktio "void Start()", joka ajetaan joka kerta kun peli aloitetaan ja "void Update()", joka ajetaan joka raamissa kerran. Pelkästään hahmon paikallaan kävelyn toteuttaminen vaatii yksinkertaisen koodin (Kuva 29). Koodiin tulee hakea animaattori yksityiseen muuttujaan. Tämä muuttuja luodaan ja aloitus funktiossa asetamme animaattorin muuttujalle. Päivitys funktiossa on määriteltävä mikä näppäin vaihtaa kävely parametrin arvoa, ja siten saa hahmon liikkeelle. Kirjoitamme funktioon if-lauseen, jossa ensin määritellään se, että kun ylös-näppäin on painettu pohjaan, kävelyparametrin suuruus tulee olla yksi. Animaattorissa määritellyt yhteydet ja niiden asetetut parametrit luetaan ja hahmo kävelee. Muulloin parametrin suuruus tulee olla nolla, jolloin hahmo on tyhjäkäynnillä.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Player : MonoBehaviour
6  {
7
8      private Animator anim;
9
10     // Use this for initialization
11     void Start()
12     {
13         anim = gameObject.GetComponentInChildren<Animator>();
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19
20         if (Input.GetKey("up"))
21         {
22             anim.SetInteger("Walk", 1);
23         }
24         else
25         {
26             anim.SetInteger("Walk", 0);
27         }
28     }
29 }
```

Kuva 29 Paikallaan kävely koodi

Jotta hahmo voi liikkua eteenpäin, tarvitsee sille luoda hahmokontrolleri. Hahmokontrolleri luodaan valitsemalla hahmo skenen hierarkiasta ja lisäämällä sille komponentti oikeasta reunasta. Hahmokontrolleri löytyy "Physics" valinnan alta. Kun hahmokontrolleri on luotu,

tulee skeneen näkyviin kapselin muotoinen häkki, tämä häkki tulee asettaa hahmon ympärille muokkaamalla sen asetuksia oikealta reunalta. Hahmokontrolleri määrittelee mikä on se osa skeneä, jota liikutetaan ohjelmakoodissa.

Ohjelmakoodissa tulee nyt luoda uusi yksityinen muuttuja hahmokontrollerille, ja pelin aloituksessa asettaa hahmokontrolleri muuttujalle. Tämän lisäksi luodaan julkinen muuttuja nopeus ja annetaan sille arvo, sekä yksityinen muuttuja liikkuvuus suunnalle. Liikkuvuussuunnan muuttuja on Vector3, joka määrittelee millä akselilla liikkuvuus tapahtuu (Unity Documentation 2018.). Kun on käytetty arvona nollaa, tarkoittaa se, ettei liikkuvuus tapahdu millään akselilla, joten tämä määritellään myöhemmin päivitys funktiossa.

Eteenpäin kävelyn lisäksi halutaan hahmon myös kääntyvän oikealle ja vasemmalle. Ohjelmakoodiin lisätään julkisia muuttujia käännösnopeudelle ja painovoimalle, sekä annetaan niille arvot.

Päivitys funktioon tulee lisätä uusi if-lause, joka testaa onko hahmo maan pinnalla. Tämän lisäksi luomme kääntö muuttujan, sekä valitsemme akselin jolla kääntö tapahtuu. Näissä lauseissa ohjelmakoodi sisäistää myös oikean ja vasemman nuolinäppäimen, sekä antaa niille akselin jolla käännetään ja kääntönopeuden. Seuraavat lauseet liikuttavat hahmokontrolleria eteen ja sivuille, sekä painovoiman mukaan maantasalle. Kokonaisuudessaan hahmon kävelyn ohjelmakoodi on yksinkertainen ja lyhyt (Kuva 30).

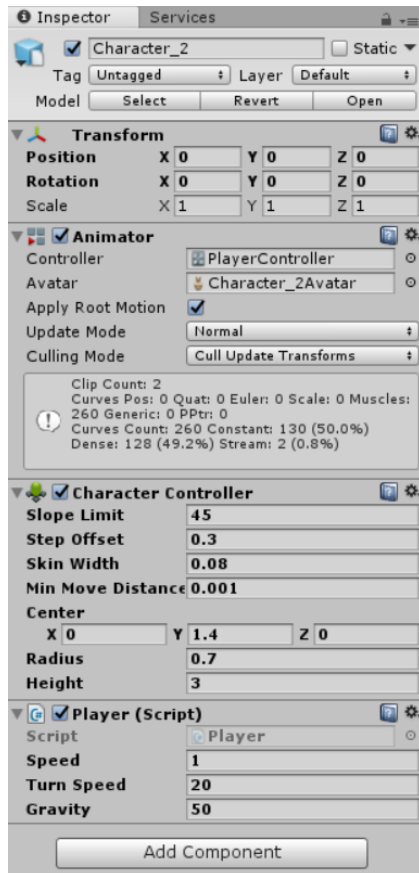
```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Player : MonoBehaviour {
6
7      private Animator anim;
8      private CharacterController controller;
9      public float speed = 1.0f;
10     public float turnSpeed = 20.0f;
11     private Vector3 moveDirection = Vector3.zero;
12     public float gravity = 50.0f;
13
14     // Use this for initialization
15     void Start () {
16         anim = gameObject.GetComponentInChildren<Animator>();
17         controller = GetComponent<CharacterController>();
18     }
19
20     // Update is called once per frame
21     void Update () {
22
23         if (Input.GetKey ("up"))
24         {
25             anim.SetInteger("Walk", 1);
26         } else
27         {
28             anim.SetInteger("Walk", 0);
29         }
30
31         if (controller.isGrounded)
32         {
33             moveDirection = transform.forward * Input.GetAxis("Vertical") * speed;
34         }
35
36         float turn = Input.GetAxis("Horizontal");
37         transform.Rotate(0, turn * turnSpeed * Time.deltaTime, 0);
38         controller.Move(moveDirection * Time.deltaTime);
39         moveDirection.y -= gravity * Time.deltaTime;
40     }
41 }

```

Kuva 30 Player.cs ohjelmakoodi kokonaisuudessaan

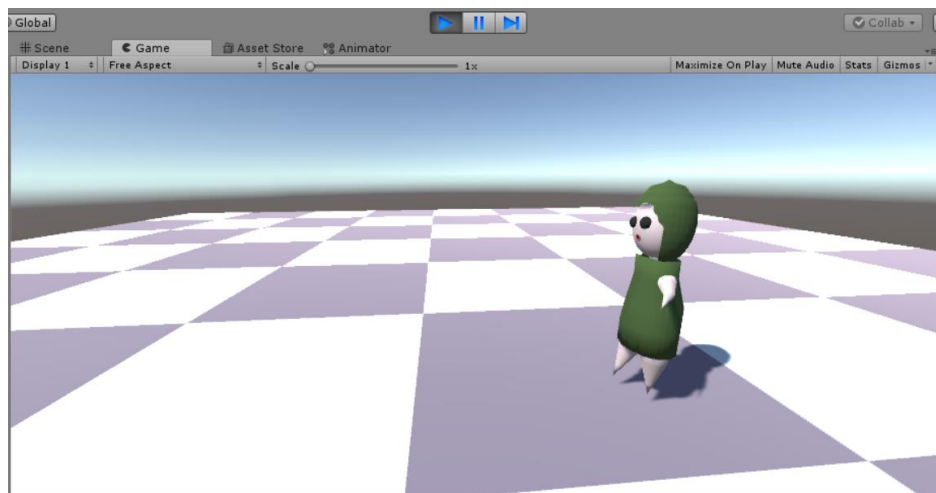
Player.cs ohjelmakoodi voidaan raahata hahmon päälle skenen hierarkiassa, jolloin Unity automaattisesti asettaa ohjelmakoodin oikealle kohdalle. Aloittaessa pelinäkömään, voidaan nyt hahmoa liikutella ylös- ja sivunäppäimillä. Ensimmäinen huomattava parannus on asettaa kamera seuraamaan hahmoa, tämän voi tehdä helposti raahaamalla skenen hierarkiassa olevan kameran hahmon päälle hierarkiassa, jolloin se sisältyy hahmon ja siten hahmokontrollerin alla oleviin liikuteltaviin objekteihin. Hahmolle tulee myös luoda alusta jolla se voi kävellä. Alusta voidaan luoda lisäämällä skeneen peliobjekti nimeltä "Plane", sekä asettaa se hahmon alle ja suurentaa sitä tarpeen mukaan. Alustalle voidaan myös asettaa materiaaleja, jotta hahmon liikkuvuus on näkyvämpää. Muita parannuksia saatamme huomata hahmon nopeudessa tai painovoimassa. Näitä on helppo muokata ilman ohjelmakoodin avaamista. Raahatessa ohjelmakoodin hahmolle, asettuu se näkyviin hahmon tietoihin josta julkisia muuttujia voidaan muokata (Kuva 31).



Kuva 31 Hahmon tiedot

8.4 Tulos

Hahmoa voidaan nyt liikuttaa painamalla näppäimistön ylös nuolta, sekä sivunuolia. Hahmo liikkuu myös taaksepäin painamalla näppäimistön alas nuolta, mutta taaksepäin kävelyllä ei ole asetettu animaatiota. Tuloksesta on luotu video, jonka voi nähdä seuraavasta linkistä: <https://youtu.be/xFs4Wv5ZNpo>.



Kuva 32 Hahmo kävelee

9 Pohdinta

3D grafiikalla on pitkä historia. Alku oli hitaampaa sekä monimutkaisempaa vaikka ensimmäiset 3D grafiikalla luodut kuvat eivät olleet kovin realistisia. Ensimmäisten 3D graafikoiden tuli oppia paljon matematiikkaa ja ohjelmistoja mallintamiseen, mutta nykyään 3D-mallintaminen on helpompaa. 3D grafiikan ala jatkaa kasvuaan ja tuo markkinoille paljon realistisia viihdetuotteita ja tuoteteollisuuden sekä lääketieteen apuvälineitä. Tulevaisuudessa virtuaalitodellisuus muuttuu yhä paremmaksi.

Markkinoilla on enenevin määrin ohjelmistoja joilla voidaan luoda 3D-mallinnettuja ja animoituja teoksia. Nykyään ohjelmistojen kehittäjät ottavat vartaasti myös huomioon monipuoliset käyttäjät sekä käyttäjäystävällisyyden myös aloittelijoille. Tämän lisäksi monet yritykset ovat luoneet ilmaisia sekä avoimen lähdekoodin tuotteita, jotta yksityiset henkilötkin voivat käyttää erinomaisia ohjelmistoja ja kerryttää taitojaan. Näistä hyvinä esimerkkeinä toimivat Blender ja Unity.

3D-mallintamisessa ja animaatiossa vaaditaan erilaisia taitoja ja tekniikoita. Yleistyvimmät tekniikat ovat rigaus tekniikka ja käänteisen kinematiikan tekniikka. Nämä mahdollistavat animoinnin paljon helpommin kuin staattisen hahmon luominen ja hahmon paloiksi paloittelu joka kerta kun halutaan sen asennon muuttuvan. Tässä opinnäytetyössä tulos tuotettiin hyödyntämällä molempia tekniikoita.

Opinnäytetyön aihe ja tuloksellisuus onnistuivat odotetusti. Alkuperäisen projektisuunnitelman mukaisesti tutkittiin 3D grafiikkaa sekä useita mallintamis ohjelmistoja ja pelimoottoreita. Tuloksen luominen 3D-mallintamalla sekä animoimisella olivat ensimmäinen vaihe joka onnistui hyvin. Seuraavana tuloksen vaiheena oli viedä valmis mallinnettu hahmo pelimoottoriin ja pelillistää se. Tuloksessa hahmo kävelee sekä esittää tyhjäkäynti animaation, joten tulos onnistui näissäkin määrin erinomaisesti.

Tulosta voitaisiin jatkokehittää luomalla sen ympärille maailma, mahdollisesti luomalla videopeli käyttämällä hahmoa, tai luomalla animaatiovideoita. Hahmoa voidaan myös kehittää huomattavasti. Hahmolle voitaisiin luoda mukautetut materiaalit ja siten oppia myös se osa-alue mallintamista. Hahmolle voidaan luoda lisää animaatioita ja lisätä ne hahmon käytettävyyteen pelimoottorissa. Tekniikoiden opettelu ja käyttäminen mahdollistavat seuraavien projektien tekemisen. Seuraavat projektit voivat olla paljon vaativampia ja keskittyä enemmän visuaaliseen mallintamiseen ja animoimiseen.

Opinnäytetyön tavoitteet täyttyivät tekijän osilta erinomaisesti. Opinnäytetyöprosessi oli aika-ajoin hidasta ja keskittyi paljon oppimiseen. Tämä kuitenkin hyödytti tekijää ja tulos on hyvä todiste ensimmäisestä tällaisesta projektista. Itse tuloksen luominen ei vienyt kovin paljoa aikaa, toki kokeilun ja erehdyksen kannalta, tulos luotiin muutamaankin otteeseen ennenkuin se oli tyydyttävä. Aihe oli erittäin mielenkiintoinen ja innosti luomaan lisää erilaisia projekteja käyttäen opittuja metodeja.

Lähteet

3D Horse. 2017. Future of 3D graphics. Luettavissa:

<https://www.3dhorse.com/blogs/3d/future-of-3d-graphics>. Luettu: 12.03.2018.

3D Horse. 2018. History of 3D Computer Graphics. Luettavissa:

<https://www.3dhorse.com/blogs/3d/history-of-3d-computer-graphics>. Luettu: 26.02.2018.

3D Horse. 2018. Uses of 3D models in today's world. Luettavissa:

<https://www.3dhorse.com/blogs/3d/uses-3d-models-today-world>. Luettu: 26.02.2018.

ArchiCGI. 2017. What is a 3D modeling? Things you've got to know nowadays.

Luettavissa: <https://archicgi.com/3d-modeling-things-youve-got-know/>. Luettu: 26.02.2018.

Blender 2.79 Manual. Importing & Exporting Files. Luettavissa:

https://docs.blender.org/manual/en/dev/data_system/files/import_export.html. Luettu: 05.05.2018.

Blender. 2018. About. Luettavissa: <https://www.blender.org/about/>. Luettu: 22.02.2018.

Blender Foundation. 2013. History. Luettavissa: <https://www.blender.org/foundation/history/>. Luettu: 22.02.2018.

Bloop. 2018. 5 Types of Animation. Luettavissa: <https://www.bloopanimation.com/types-of-animation/>. Luettu: 20.05.2018.

CAD/CAM Services. 2018. 8 Industries that 3D modeling has revolutionized. Luettavissa:

<https://www.cadcam.org/blog/8-industries-3d-modeling-revolutionized/>. Luettu: 22.02.2018.

Lile, D. 2015. Blender Character to Unity part 1 of 2. Katsottavissa:

<https://www.youtube.com/watch?v=h8ol0n5kAlg>. Katsottu: 06.05.2018.

Lile, D. 2015. Blender Character to Unity part 2 of 2. Katsottavissa:

<https://www.youtube.com/watch?v=z6h3aE9LlXk>. Katsottu: 06.05.2018.

Dunietz, J. 2016. The most important object in computer graphics history is this teapot. Luettavissa: <http://nautil.us/blog/the-most-important-object-in-computer-graphics-history-is-this-teapot>. Luettu: 25.02.2018.

Elhady, H. 2017. Top game engines in 2018. Instabug. Luettavissa: <https://blog.instabug.com/2017/12/game-engines/>. Luettu: 13.03.2018.

Forbes. 2016. 3D Animation in film will always be far ahead of video games. Luettavissa: <https://www.forbes.com/sites/quora/2016/10/13/3d-animation-in-film-will-always-be-far-ahead-of-video-games/#7846228433fb>. Luettu: 13.03.2018.

Gamasutra. 2010. Press Release: Unity Technologies Celebrates Five Years of Continual Leadership and Innovation in Making Cutting Edge Game Technology. Luettavissa: https://www.gamasutra.com/view/pressreleases/140640/Unity_Technologies_Celebrates_Five_Years_of_ContinualLeadership_and_Innovation_in_Making_Cutting_Edge_GameTechnology.php. Luettu: 22.02.2018.

GameDesigning. 2018. The top 10 video game engines. Luettavissa: <https://www.gamedesigning.org/career/video-game-engines/>. Luettu: 13.03.2018.

Lacoma, T. 2017. Get your next project started right with the best 3D modeling software. Digital Trends. Luettavissa: <https://www.digitaltrends.com/computing/best-3d-modeling-software/>. Luettu: 12.03.2018.

Pitzel, S. 2017. Character Animation: Skeletons and Inverse Kinematics. Intel Game Dev. Luettavissa: <https://software.intel.com/en-us/articles/character-animation-skeletons-and-inverse-kinematics>. Luettu: 06.03.2018.

Pluralsight. 2014. How animation for games is different from animation for movies. Luettavissa: <https://www.pluralsight.com/blog/film-games/how-animation-for-games-is-different-from-animation-for-movies>. Luettu: 13.03.2018.

Pluralsight. 2015. How to Create Your First Character Rig in Blender: Part 1 - Setting up the Armature. Luettavissa: <https://www.pluralsight.com/blog/tutorials/how-to-create-your-first-character-rig-in-blender-part-1>. Luettu: 21.05.2018.

Lague, S. 2012. Blender Tutorial: Basic Walk Cycle. YouTube. Katsottavissa: <https://www.youtube.com/watch?v=d-wQ8nRWTBs>. Katsottu: 22.05.2018.

Lague, S. 2013. Blender Tutorial: Basics of Character Rigging. YouTube. Katsottavissa: <https://www.youtube.com/watch?v=cGvalWG8HBU>. Katsottu: 21.05.2018.

Slick, J. 2017. 3D defined – What is 3D? Lifewire. Luettavissa: <https://www.lifewire.com/what-is-3d-1951>. Luettu: 06.03.2018.

Slick, J. 2017. The best free 3D software to download. Lifewire. Luettavissa: <https://www.lifewire.com/free-3d-software-list-2005>. Luettu: 06.03.2018.

Slick, J. 2018. What is 3D modeling? Lifewire. Luettavissa: <https://www.lifewire.com/what-is-3d-modeling-2164>. Luettu: 06.03.2018.

Slick, J. 2018. What is Rigging in 3D Animation? Lifewire. Luettavissa: <https://www.lifewire.com/what-is-rigging-2095>. Luettu: 06.03.2018.

Surfaced Studio. 2017. Blender for Absolute Beginners. Luettavissa: <https://www.surfacedstudio.com/tutorials/blender-for-absolute-beginners>. Luettu: 18.05.2018.

Unity Documentation. 2017. Importing Objects from Blender. Luettavissa: <https://docs.unity3d.com/560/Documentation/Manual/HOWTO-ImportObjectBlender.html>. Luettu: 05.05.2018.

Unity Documentation. 2017. Rig Tab. Luettavissa: <https://docs.unity3d.com/Manual/FBXImporter-Rig.html>. Luettu: 05.05.2018.

Unity Documentation. 2018. Unity User Manual (2018.1). Luettavissa: <https://docs.unity3d.com/Manual/UnityManual.html>. Luettu: 20.05.2018.

Unity Documentation. 2018. Vector3. Luettavissa: <https://docs.unity3d.com/ScriptReference/Vector3.html>. Luettu: 14.05.2018.

Unity Technologies. 2018. Store. Luettavissa: https://store.unity.com/?_ga=2.126188170.667900742.1520429600-1125029920.1519318069. Luettu: 22.02.2018.

Unity Technologies. 2018. About Us. Luettavissa: <https://unity3d.com/public-relations>. Luettu: 22.02.2018.

